

MODELADO, ANÁLISIS, IDENTIFICACIÓN Y CONTROL DE SISTEMAS MULTIVARIABLES

Antonio Sala
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Recopilación de documentos y material audiovisual

VERSIÓN DEL DOCUMENTO:

Fecha y hora: 12/12/2025 [11:07]
Núm. objetos de aprendizaje: 972
Tiempo total lecciones en video: 208:42:17
[propio (939): 203:28:56, terceros (33): 05:13:21]

Website última versión: [HTML] [PDF]

Publico aproximadamente un vídeo por semana... Actualiza si el documento tiene más de 2 meses.

©2025, Antonio Sala Piqueras, DISA-UPV.

Suscríbete en 1 click a canales **YouTube** asociados: [@asalacontrol, Español] [@ASalaControlEN, English]

Capítulo I

Presentación

Este documento presenta enlaces a múltiples materiales on-line elaborados por mí (y por terceros) que permitirán al lector interesado profundizar en la temática de modelado y control multivariable (fundamentalmente enfocado al control lineal y control robusto de procesos lineales inciertos) y el análisis multivariante de conjuntos de datos para identificar modelos estáticos o dinámicos que los relacionan. De todas formas, algunos materiales también abarcan conceptos de modelado, linealización, respuesta en frecuencia, etc. propios de asignaturas iniciales del área de Automática en estudios de ingeniería.

Existen abundantes materiales y vídeos sobre los temas aquí discutidos en lengua inglesa, pero no tantos en lengua castellana, y menos en algunos de los conceptos más avanzados que aquí se tratan. Además, se ha tratado de ordenar, relacionar y presentar un enfoque unificado que facilite su seguimiento. Por ello, considero que la información aquí recopilada es relevante para ayudar en estudios de Grado, Máster y Doctorado en Ingeniería de Sistemas y Automática, en el contexto universitario español y latinoamericano.

Es importante recalcar, no obstante, que (a) un alumno de Grado necesitará bastante material introductorio adicional, (b) un alumno de especialidades *no* relacionadas con el área de Automática posiblemente busque exposiciones más concisas, pocas fórmulas, y *recetas* prácticas sin necesidad de comprender en profundidad el porqué de casi todo, y (c) un estudiante de Doctorado debería consultar referencias bibliográficas recientes para comprender el estado del arte actual, que supera y refina muchísimos de los conceptos aquí discutidos, que en su mayoría datan del período 1950–2000.

También se debe hacer notar que el énfasis está puesto en los conceptos e ideas de aplicación en Ingeniería y, por tanto, contienen simplificaciones, generalizaciones y omisiones que adolecen de defectos en formalidad, abstracción o modernidad del tratamiento; por ello, posiblemente estos materiales no sean los más indicados para alguien que aborde el análisis de datos y control desde una perspectiva de, por ejemplo, un estudiante de grado/máster en ramas de las Ciencias Matemáticas.

Aunque para un estudiante avanzado (etapa final de Máster, Doctorado) podrían ser materiales adecuados para un estudio autodidacta, tienen el inconveniente que, por una parte, abordan muchos temas diferentes y, por otra, existen “discontinuidades” en la exposición (faltan materiales teóricos que deberían estar entre algunos de los contenidos) y no contienen todos los ejemplos, ejercicios, prácticas, problemas a resolver, trabajos, etc. que serían aconsejables en una asignatura universitaria completa. Probablemente, estaría más indicado que los contenidos aquí enlazados (un subconjunto de los mismos, seguramente) fueran utilizados como material de apoyo y que el estudiante tuviera un profesor (o tutor/director de tesis) que le recomendara qué presentaciones revisar (y en qué orden), qué materiales adicionales revisar para completar conceptos esenciales que aquí no están desarrollados, y al que preguntar dudas o con el que discutir ejercicios y problemas que éste propusiera, por ejemplo, en un entorno de tipo *clase inversa* (*flipped classroom*).

Como reflexión final, espero que, a los que estéis leyendo esto, os sea útil y os motive a profundizar en vuestro conocimiento de esta interesante disciplina.

Antonio Sala

Departamento de Ingeniería de Sistemas y Automática
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Nota sobre originalidad de los contenidos

Estos materiales abordan contenidos “asentados” en la comunidad científica, muchos de ellos ya “clásicos”, desarrollados hace veinte o más años. La mayoría de las ideas y algoritmos son, por tanto, bien conocidos entre los expertos de la especialidad y, por tanto, no se reivindica prácticamente ninguna originalidad a nivel de “investigación” sobre los mismos.

La única “originalidad” es en el ámbito “didáctico”: el criterio de selección y priorización de los conceptos, la organización de los materiales y la forma de presentarlos en las transparencias y ejemplos aquí proporcionados. Por ello, este documento o los enlazados en él no deberían ser citados en un contexto de “investigación” como una fuente autoritativa de recomendaciones de ingeniería o resultados/demostraciones matemáticas, dado que no son originales/seminales, y consideraciones expositivas pensando en el público objetivo (estudiantes) hacen que, intencionalmente, haya simplificaciones, variaciones, o aserciones no del todo justificadas.

Para evitar distraer la atención del lector, se ha evitado la profusión de citas que es habitual en los textos de investigación. No obstante, se afirma aquí explícitamente que existen abundantes monografías (en lengua inglesa, principalmente) que desarrollan (en muchos casos, con mucha más profundidad y formalismo) la práctica totalidad de los conceptos aquí discutidos. Una vez familiarizado con las ideas básicas, el estudiante que desee mayor detalle de las ideas en los capítulos de este texto debería consultar a su profesor o supervisor de TFM/Tesis, para que le remita a las referencias autoritativas o contribuciones seminales correspondientes.

Derechos de utilización

El material en YouTube está sujeto a la Licencia de YouTube estándar. Las condiciones legales para usar/enlazar a los contenidos alojados en media.upv.es pueden consultarse en <https://media.upv.es/#/terminosdeuso>. La licencia de uso de contenidos en riunet.upv.es aparece en cada uno de los elementos allí alojados; en su defecto, si no aparece, se supone CC:by-nc-nd.

Se entiende que contenidos de terceros aquí incluidos, provenientes de enlaces públicos, no infringen el *copyright* de los mismos. Si alguno de los autores desea que se elimine la referencia a sus materiales, que me lo comunique y haré efectiva dicha eliminación. La copia o redistribución de contenidos de terceros deberá respetar los derechos de uso que sus autores hayan establecido.

En cuanto al material de mi autoría, se reservan todos los derechos de uso, copia, y propiedad intelectual de los materiales en este documento y los materiales audiovisuales/PDF/código enlazados desde el mismo, tanto respecto a su uso *verbatim* o como base de obras derivadas. Se exceptúa de esta reserva y se autoriza expresamente el uso para *estudio personal*, de alumnado y profesorado de universidades *públicas* en títulos *oficiales* y la recomendación por parte de sus profesores de este documento. En otros contextos (cursos remunerados, academias privadas, publicaciones, ...), contácteme si piensa obtener *dinero* o *mérito* con esto u obras derivadas y negociaremos la adecuada cesión, no problem.

Tabla de Contenidos

I	Presentación	1
II	Organización y uso	11
II.1	Organización del material	11
II.1.1	Otros materiales/cursos online alternativos	15
I	Modelado y Análisis del Comportamiento Dinámico de Sistemas	16
1	Modelado de sistemas físicos	17
1.1	Señales y sistemas	17
1.1.1	Concepto de “estado” en sistemas físicos	19
1.2	Modelos físicos en forma de ecuaciones diferenciales	20
1.2.1	Conceptos básicos	21
1.2.2	Modelos algebraico-diferenciales generales (índice 1)	24
1.2.3	Más ejemplos	24
1.2.4	Caso de estudio sistema térmico (calentador de líquido)	26
1.2.5	Modelos algebraico-diferenciales de índice superior	28
1.3	Autoevaluación	30
2	Simulación de sistemas y casos de estudio adicionales	31
2.1	Simulación (integración numérica)	31
2.2	Casos de estudio de modelado de sistemas mecánicos	37
2.2.1	Caso estudio: simulación interacción gravitatoria entre varios cuerpos	37
2.2.2	Mecanismo piñón-cremallera	38
2.2.3	2 masas y muelle, mediante Euler-Lagrange	39
2.2.4	Movimiento lateral de un tiovivo	39
2.2.5	Dinámica longitudinal de vuelo de aeronave (fugoide)	41
2.2.6	Caso de estudio: móvil restringido en un alambre	42
2.2.7	Sistema Bola-carrito de 2 grados de libertad	45
2.2.8	Sistema catapulta barra-bola	46
2.2.9	Péndulo de múltiples eslabones	47
2.3	Modelado basado en estadística: ejemplo biorreactor	48
2.4	Modelos de parámetros distribuidos (medio continuo)	48
3	Linealización	51
3.1	Funciones de una variable	51
3.2	Aproximación de funciones de varias variables	52
3.2.1	Ecuaciones de sistemas dinámicos (linealización de EDO)	53
3.3	Ejemplos adicionales	54
3.4	Autoevaluación	58

4	Respuesta temporal	60
4.1	Transformada de Laplace	60
4.1.1	Función de transferencia (SISO), Matriz de transferencia (MIMO)	60
4.2	Respuesta temporal de sistemas sencillos	61
4.2.1	Caso de estudio masa-muelle-amortiguador	61
4.2.2	Sistemas genéricos	64
4.2.3	Otros ejemplos:	66
4.2.4	Sistemas con retardo en dinámica, o entradas “por tramos”	68
4.2.5	Matriz de transferencia (MIMO)	74
4.3	Teorema del Valor Final	75
4.4	Recapitulación: respuesta de sistemas sencillos ante entradas típicas	76
4.4.1	Sistemas de primer orden (+retardo)	76
4.4.2	Sistemas de segundo orden	78
4.4.3	Efecto de los ceros en la dinámica de los sistemas	79
4.4.4	Identificación experimental basada en respuesta ante entradas típicas	84
4.5	Respuesta temporal en representación interna	87
4.5.1	Respuesta libre	87
4.5.2	Régimen Estacionario	88
4.6	Ejemplos de modelado y cálculo de respuesta temporal	88
4.6.1	Caso de estudio: motor de corriente continua con carga mecánica	90
4.6.2	Caso de estudio: calentamiento pieza de dos capas	93
4.6.3	Caso de estudio: transformador eléctrico	95
4.6.4	Caso de estudio: calentador tubular	97
4.6.5	Sistemas con retardo	100
5	Respuesta en frecuencia de sistemas lineales	102
5.1	Fórmulas y diagramas de respuesta en frecuencia	102
5.2	Filtros analógicos elementales	104
5.2.1	Ejemplos	104
5.3	Transformada de Fourier	105
6	Sistemas de tiempo discreto: representación y análisis	107
6.1	Respuesta temporal	107
6.2	Respuesta en frecuencia	107
7	Procesado estadístico de señal: procesos estocásticos	108
7.1	Procesos estocásticos: definiciones básicas y ejemplos	108
7.1.1	Descripción frecuencial de los procesos estacionarios	110
7.1.2	Generación de muestras de un proceso estocástico gaussiano	111
7.1.3	Procesos estocásticos de varias salidas (MIMO)	112
7.1.4	Componentes principales (transformación autofunciones Karhunen-Loève)	113
7.2	Ecuaciones en diferencias estocásticas: Respuesta ante entrada aleatoria de sistemas discretos	116
7.2.1	Ejemplos	116
7.3	Ecuaciones diferenciales estocásticas (procesos estoc. en tiempo continuo)	118
7.3.1	El proceso de Wiener (movimiento Browniano)	118
7.3.2	Ecuaciones Diferenciales Estocásticas	119
7.3.3	EDO Estocásticas lineales	120
7.3.4	Solución exponencial de ecuaciones de medias, varianzas y autocovarianzas	122
7.4	Estimación en series temporales a partir de Kernel de covarianza	125
7.4.1	Estimación con medidas de “velocidad” (derivadas parciales de proceso estocástico)	127
7.4.2	Ejemplo: el proceso (kernel) tipo Matérn	128
7.4.3	Procesos estocásticos bidimensionales	129
7.5	Estimación a partir de EDO estocástica, ejemplo	130
7.6	Optimización Bayesiana	132
7.6.1	Motivación y esbozo de la metodología	132
7.6.2	Detalles, código y ejemplos	134

8	Sistemas multivariables: análisis de propiedades	137
8.1	Causalidad y realizabilidad	137
8.2	Estabilidad, ganancia, polos y ceros, frecuencia	137
8.2.1	Ejemplos Matlab	139
8.3	Controlabilidad y Observabilidad	142
8.4	Manipulación e interconexión de sistemas	144
8.4.1	Estabilidad interna, controlabilidad y observabilidad en bucle cerrado	145
9	Aproximación de orden reducido de sistemas lineales	147
9.1	Caso monovariable, ideas básicas	147
9.2	Caso multivariable: planteamiento	149
9.3	Reducción modal	149
9.4	Reducción equilibrada	150
9.5	Ejemplos de código	150
II	Control Monovariable	153
10	Reguladores monovariantes en tiempo continuo	154
10.1	Definiciones básicas de sistemas de control	154
10.2	Control en Bucle Abierto	155
10.3	Control Todo-Nada (On-Off)	156
10.4	Reguladores PID: Comprensión “intuitiva” de las acciones básicas de control	157
10.5	Demostraciones PID en equipos reales (de terceros)	161
10.6	Control PID: metodologías teóricas de diseño y ejemplos Matlab	162
10.7	Error en régimen estacionario en bucle cerrado	177
10.8	Análisis de las prestaciones de un bucle de control: respuesta en frecuencia	179
10.8.1	Análisis en frecuencia de bucles de control: criterio de estabilidad de Nyquist	179
10.8.2	Análisis en frecuencia de bucles de control: respuesta en frecuencia en bucle cerrado	181
11	Control por computador: selección de período de muestro y discretización	183
11.1	Muestreo y reconstrucción de señales analógicas	184
11.2	Control por computador: problema a resolver	184
11.3	Discretización de procesos lineales (para diseño posterior de reguladores discretos)	185
11.3.1	Discretización exacta de efecto acción de control (zoh)	185
11.3.2	Discretización por interpolación de perturbaciones con retenedor orden 1	186
11.4	Discretización aproximada de filtros y reguladores continuos	186
11.4.1	Discretización (aproximada) de reguladores en representación interna	188
11.5	Diseño de reguladores discretos por “continuización” con garantía de estabilidad	189
11.6	Selección de período de muestreo en aplicaciones de control	190
11.6.1	Selección de período de muestreo para discretización de reguladores continuos	191
11.6.2	Caso de estudio, simulación sampled-data	193
12	Control por modelo interno (IMC)	194
12.1	Idea general y metodología para plantas estables	194
12.1.1	IMC en sistemas de fase no mínima	197
12.2	Sistemas con retardo: predictor de Smith	199
12.3	Diseño de PID's con IMC	199
12.4	Ejercicios adicionales	203
12.4.1	Antiwindup	205
12.4.2	Diseño IMC para implementación por computador (tiempo discreto)	205
12.5	Caso general inestable: parametrización de Youla-Kucera	206

III Control Multivariable	208
13 Control de procesos complejos: etapas iniciales	209
13.1 Problemas a resolver, selección de punto de operación	209
13.2 Selección de actuadores y variables controladas primarias	211
13.2.1 Ejemplos adicionales	216
13.3 Aplicación a robótica: elipsoides de manipulabilidad y fuerza	224
13.4 Selección de la estructura de control	226
14 Control multivariable: estructura descentralizada	228
14.1 Control multibucle	228
14.1.1 Consideraciones adicionales sobre la matriz RGA	230
14.2 Prealimentación (Feedforward)	231
14.2.1 Ejemplos y casos de estudio	232
14.3 Desacoplamiento	235
14.3.1 Caso de estudio: tanque de mezclado	237
14.3.2 Desacoplamiento SVD	237
14.3.3 Explicación más detallada, comprensión “intuitiva”	238
14.4 Desacoplamiento/linealización por realimentación del estado	242
14.4.1 Robótica móvil: control por punto descentrado, caso de estudio	244
14.5 Control en cascada	246
14.6 Estructuras mixtas y otras propuestas	246
14.7 Control con dos grados de libertad	249
14.8 PID avanzado: antiwindup por modo de seguimiento (tracking mode)	250
14.9 Casos de estudio y ejemplos adicionales	252
14.9.1 Unificación de resultados	252
14.9.2 Casos de estudio de procesos industriales	253
14.9.3 Tanque de mezclado (V2) con concentración no homogénea y retardo	257
14.9.4 Otros ejemplos	264
14.10 RESUMEN: visión general del control ‘usual’ en industria	264
15 Control centralizado: asignación de polos	267
15.1 Realimentación del estado	267
15.2 Estimación del estado: Observadores	267
15.2.1 Observador de orden reducido	269
15.3 Realimentación de la salida	269
15.4 Caso de estudio: sistema mecánico no lineal de cuarto orden	270
15.5 Doble integrador	270
15.5.1 sistema mecánico no lineal de cuarto orden	271
15.6 Rechazo de perturbaciones	271
15.6.1 Perturbaciones constantes: acción integral	272
15.6.2 Caso general: perturbaciones deterministas constantes, rampa, senoidales	273
16 Control óptimo	276
16.1 Introducción al control basado en optimización	276
16.2 Sintonizado de reguladores PID basado en optimización	277
16.2.1 Simulink Response Optimization	277
16.2.2 Control óptimo via ode45+fminunc (ejemplo Matlab)	281
16.3 Control multivariable LQR	284
16.4 Control Predictivo (enlaces a terceros)	284
16.5 Control óptimo no lineal (enlaces a terceros)	284
16.6 Programación dinámica (En elaboración)	285

17 Predicción óptima en procesos con perturbaciones aleatorias (filtro de Kalman)	286
17.1 Preliminares	286
17.2 El filtro de Bayes recursivo (motivación y ejemplo ‘tigre oculto’)	287
17.3 Predicción óptima en procesos gaussianos en tiempo discreto	288
17.3.1 Predicción óptima no causal: suavizado RTS	290
17.4 Casos de estudio Kalman/RTS	291
17.4.1 Caso 1: estimación de fuerza aplicada sobre un sistema masa-muelle	291
17.4.2 Caso 2: eliminación de tendencias (rampa)	291
17.4.3 Caso 3: ejemplo donde NO funciona bien	293
17.5 Estimación aproximada en procesos no lineales	293
17.5.1 Filtro de Kalman extendido	294
17.5.2 Filtro de Kalman-Uhlmann “unscented”	296
17.5.3 Filtro de Partículas	298
18 Estimación y control: escenarios avanzados de muestreo y muestreo no convencional	300
18.1 Preliminares al control con muestreo no convencional	300
18.2 Control bi-frecuencia (dual rate) por asignación de polos	301
18.3 Estimación/filtrado con muestras no equiespaciadas	301
18.3.1 Caso de estudio: fusión de sensores a diferente período con filtro de Kalman	302
19 Control óptimo \mathcal{H}_2 y \mathcal{H}_∞	303
19.1 Preliminares: normas de sistemas	303
19.1.1 Interpretación de normas \mathcal{H}_2 y \mathcal{H}_∞ : ejemplos Matlab	303
19.1.2 Cálculo de la norma \mathcal{H}_2 de sistemas lineales	304
19.1.3 Cálculo de la norma \mathcal{H}_∞ de sistemas lineales	304
19.2 Planta generalizada	305
19.2.1 Planta generalizada ponderada para control óptimo	308
19.3 Ejemplos de código	309
19.3.1 Caso de estudio: compensación perturbaciones entrada	311
19.3.2 Selección de pesos con plantas previamente escaladas	312
19.3.3 Ejemplo de aplicación: control en cascada (sensor extra)	313
19.3.4 Ejemplo de aplicación: control en cascada con actuador extra	314
19.3.5 Prealimentación (feedforward)	315
19.3.6 Ejemplo de aplicación: control con 2 grados de libertad	316
19.3.7 Ejemplos: control de sistemas térmicos	317
19.3.8 Observadores estacionarios \mathcal{H}_2 y \mathcal{H}_∞	319
19.4 Limitaciones de prestaciones en procesos inestables o de fase no mínima	321
19.5 Antiwindup en controladores en representación interna	322
19.5.1 Caso de estudio: implementación en un regulador \mathcal{H}_∞ discreto	323
19.6 Conclusiones	323
IV Análisis multivariante, Identificación y Detección de Cambios	324
20 Identificación/ajuste de modelos (sin dinámica)	325
20.1 Mínimos cuadrados (caso estático)	326
20.2 Mínimos cuadrados recursivos	328
20.2.1 Algoritmos recursivos sin olvido	328
20.2.2 Algoritmos con olvido (factor de olvido / filtro de Kalman)	329
20.3 Mínimos cuadrados totales	330
20.3.1 Ejemplos Matlab	331
20.4 Otros índices de coste para regresión	332
20.5 Métodos Kernel	332
20.5.1 Regresión mínimos cuadrados contraída (ridge-regression), versión Kernel	333
20.5.2 Regresión semiparamétrica (kernel+modelo con parámetros explícitos)	335

21 Análisis multivariante de datos	337
21.1 Análisis de componentes principales	337
21.1.1 Casos de estudio	338
21.1.2 Componentes principales, versión Kernel (K-PCA)	341
21.2 Regresión, correlaciones canónicas	341
21.2.1 PCR: regresión por componentes principales	341
21.2.2 PLS: Mínimos cuadrados parciales	342
21.2.3 CVA: análisis de variables canónicas	343
21.2.4 Ejemplos de código	345
21.3 Extensión al caso no lineal	345
21.4 Modelos para clasificación [introducción al problema]	346
22 Identificación de Sistemas Dinámicos	348
22.1 Modelos polinomiales discretos ARX/OE	348
22.1.1 Ejemplos Matlab	348
22.1.2 Identificación de series temporales sin entrada	351
22.1.3 Regularización con métodos Kernel (ident. rep. impulso)	353
22.2 Identificación de sistemas en tiempo continuo	355
22.3 Identificación directa de parámetros físicos	355
22.3.1 Identificación no lineal con filtro de Kalman extendido	358
22.4 Consideraciones prácticas	359
23 Sistemas dinámicos lineales en representación interna: identificación subespacio	360
23.0.1 Ejemplo motivador: PCA de datos extendidos (lifting)	360
23.1 Identificación de múltiples salidas futuras (control predictivo)	361
23.2 Algoritmos de Identificación Subespacio	362
23.2.1 Ejemplo de implementación en Matlab	363
24 Detección de anomalías (ideas preliminares)	366
24.1 Test estadístico de hipótesis (media y varianza, monovariable, χ^2 , t -Student)	366
24.2 Test estadístico de hipótesis: caso multivariable χ^2 , T^2	368
24.3 Test estadísticos de detección de anomalías sobre resultados de PCA	370
V Control Robusto	372
25 Motivación e ideas preliminares	373
25.1 Introducción y planteamiento del problema	373
25.2 Compromiso especificaciones-robustez	373
25.2.1 Condicionamiento numérico	376
25.3 Diseño de controladores robustos mediante optimización minimax genérica	377
26 Análisis de estabilidad robusta	379
26.1 Los inicios (1940-60): márgenes de estabilidad asociados a criterio de Nyquist	379
26.1.1 Inconvenientes de los márgenes de fase/ganancia/retardo	380
26.1.2 Caso de estudio	381
26.2 Teorema de pequeña ganancia	381
26.3 Análisis de estabilidad robusta mediante pequeña ganancia	382
26.3.1 Incertidumbre aditiva	382
26.3.2 Otras incertidumbres no estructuradas (multiplicativa, coprime, ...)	384
26.3.3 Caso de Estudio	387

27 Diseño de controladores robustos ante incertidumbre no estructurada	388
27.1 Sensibilidad mixta	388
27.1.1 Ejemplos de código	390
27.1.2 Problemas debido a cancelaciones polo/cero y escalado	392
27.2 Control óptimo ante incertidumbre coprima normalizada, metodología Glover-McFarlane	394
27.2.1 Ejemplos de código	395
27.3 Caso de estudio: doble integrador	397
28 Prestaciones robustas e incertidumbre estructurada	399
28.1 Prestaciones robustas, pequeña ganancia escalado	399
28.1.1 Ejemplos de código	400
28.2 Incertidumbre estructurada y metodologías iterativas	401
28.3 Uso de la Robust Control Toolbox de Matlab, caso de estudio 1	402
28.3.1 Modelado	402
28.3.2 Control Robusto	403
28.3.3 Análisis y diseño de PID's	405
28.3.4 Caso multivariable	405
28.3.5 Significado de márgenes de robustez	406
28.4 Caso de estudio 2 prestaciones robustas	407
28.5 Caso de estudio 3, observador con musyn	410
29 Modelado de sistemas inciertos/no lineales para control robusto	412
29.1 Modelado Linear-Fractional-Transformation	412
29.2 Sistemas no lineales: acotación de sector	414
29.2.1 Ejemplos Matlab	415
29.3 Linealización alrededor de trayectorias	417
30 Pasividad	419
30.1 Definiciones básicas y utilidad	419
30.1.1 Refinamientos, exceso de pasividad	420
30.1.2 Sistemas lineales pasivos: condiciones respuesta frecuencia	421
30.1.3 Exceso de pasividad, criterio del círculo (Matlab)	422
30.2 Pasividad para control robusto \mathcal{H}_∞	423
30.2.1 Transformaciones pasivo-contractivo	423
30.2.2 Caso de estudio (planta generalizada)	424
31 El enfoque LMI a control robusto y planificación de ganancia (DRAFT)	426
31.1 Problemas geométricos resolubles con LMIs	426
31.1.1 Condicionamiento mínimo	426
31.1.2 Geometría de elipses y elipsoides	427
31.1.3 Conjuntos representables LMI/SDP	429
31.1.4 Distancia Mínima entre elipses (o, en general, LMI sets)	430
31.2 Control de coste garantizado	431
Apéndice	436
A Matemáticas para control multivariable	436
A.1 Matrices, geometría	436
A.1.1 Elipses y Elipsoides	438
A.2 Optimización	440
A.2.1 Mínimos cuadrados	441
A.3 Variable Compleja	442
A.4 Descomposición en valores singulares (SVD)	442

B	Introducción a Estadística para ingenieros de control y procesos	445
B.1	Motivación	445
B.2	Conceptos básicos: variables aleatorias, probabilidad, media, varianza	446
B.3	Variables aleatorias: caso multivariable	447
B.3.1	Transformación (función) de una variable aleatoria	447
B.3.2	Variables aleatorias multidimensionales	448
B.4	Intervalos (monovariable) y elipsoides (multivariable, normal) de confianza	454
B.5	Predicción estadística	457
B.5.1	Interpretaciones deterministas y estadísticas de las soluciones del problema de mínimos cuadrados	459

Capítulo II

Organización y uso

II.1 Organización del material

En vez de ser un “libro de texto” clásico, este documento es una recopilación de presentaciones y vídeos explicativos de las mismas. Cada uno de los vídeos dispone de una breve descripción de su contenido y un enlace donde ser visionado.

Se recomienda resolución 1080p o superior en monitores o TVs en conexiones con ancho de banda suficiente, dado que la claridad del texto a pantalla completa mejora significativamente.

Se recomienda usar los controles de velocidad de reproducción para, por un lado, *ralentizar* el audio y el vídeo para su mejor seguimiento cuando se estudie en profundidad (una reproducción al 90% sería aconsejable en muchos casos) o, por otro lado, para *acelerarlo* si sólo se desea un “vistazo rápido” a conceptos ya conocidos. Dependiendo de la configuración del navegador, podría ser conveniente habilitar el permiso de “*reproducción automática*” del mismo.

Otro enlace activo a la URL de materiales (PDF, Matlab) asociados al vídeo también estará disponible para aquellas presentaciones donde se disponga de ellos.

Los títulos de la tabla de contenidos y las referencias cruzadas también son enlaces activos que remiten a la página correspondiente de este documento.

Nota: El tamaño de fuente (especialmente de ejemplos de código) puede dificultar la lectura de los materiales presentados en vídeo en pantallas de pequeño tamaño (smartphone).

Perfil del lector

Estos materiales están pensados, en principio, para alumnado de la especialización en *Control de procesos, automatización y robótica* del *Master Universitario en Ingeniería Industrial* de la *Escuela Técnica Superior de Ingenieros Industriales* (ETSII-UPV) de la que soy profesor, y también para asentar conceptos de interés a alumnos que se inician en Doctorado del área de Automática.

Históricamente, su contenido “central” se gestó inicialmente con materiales de asignaturas de *Ingeniería de Control* y de *Identificación y Control de Sistemas Complejos*; posteriormente, se han añadido con algunos materiales de *Control por computador* y otros de nivel más introductorio (*Sistemas Automáticos*), o explicaciones complementarias salidas de dudas o discusiones con alumnos de TFM/Doctorado, o de seminarios impartidos por mí en el pasado.

Materiales incompletos: El contenido de este documento no está “completo” (ni lo estará): es un trabajo en elaboración, y las disciplinas de modelado, identificación y control multivariable son muy amplias y relevantes en la ingeniería de la rama industrial. Existen muchos conceptos clave en control que *no* están presentes y, por tanto, existen “discontinuidades” en la exposición de los conceptos.

Cada profesor y cada estudiante tienen unos intereses diferentes y su opinión sobre qué se ha escogido, cómo se ha presentado, el orden de los temas o su relevancia será, con seguridad, distinta a la mía.

Nota: dado que este material está “en elaboración continua”, es posible que alguno de los enlaces externos cambie de destino, de nombre, o sea eliminado. Disculpad las molestias si accedéis a este site (o a una versión PDF no actualizada) dentro de un tiempo y os encontráis con dicha situación.

Vídeos privados: en algunos casos, existen vídeos no disponibles (“privados”). Están pendientes de edición o verificación por mi parte, o bien están en cola para su próxima publicación. Vuelve, pues, a intentarlo más tarde o suscríbete para ser notificado de la publicación.

Vídeos en inglés: En la colección aparecen algunos vídeos en idioma inglés. Se trata bien de materiales de terceros autores, o bien de vídeos míos ya publicados en dicho idioma cuya versión en español está pendiente de grabación/publicación. A su debido tiempo se publicará la versión en español; se enlaza al inglés de forma temporal.




Simbología de niveles de dificultad

Los materiales están adaptados a diferentes niveles de dificultad y perfil del alumno que los visiona. La dificultad está indicada con estrellas en cada uno de los vídeos, simbolizando lo siguiente:

Símbolo	Dificultad:
*	Sencilla, nivel introductorio o divulgativo. Comprensible para estudiantes que hayan superado las materias básicas de primeros cursos de enseñanzas técnicas y estén interesados en iniciarse en asignaturas del área de Automática.
**	Contenidos sencillos, o fundamentos básicos del área de Automática. Aborda conceptos relacionados con los contenidos de la primera o segunda asignatura (grado) en el área.
***	Contenidos para alumnos que ya hayan adquirido las competencias básicas del área: casos de estudio de mayor entidad para fase final de asignaturas básicas de grado, o para estudiantes que quieren profundizar en determinados aspectos (grado avanzado, optatividad, asignaturas iniciales de Máster).
****	Contenidos de mayor complejidad, fundamentalmente de interés para estudiantes siguiendo una especialización/intensificación en Automática (p. ej., Máster).
*****	Contenidos más avanzados y demostraciones matemáticas complicadas (alumnos de especialización Máster, TFM, Doctorado).

Relevancia, importancia de cada contenido

También está indicada en los vídeos, junto a la “dificultad”, su relevancia (importancia) didáctica para un estudiante de Grado o Master en alguna especialidad de ingeniería de la rama industrial, como sigue:





Símbolo	Relevancia:
	<i>Prioridad alta:</i> conceptos teórico/prácticos importantes, que no deben ser pasados por alto para comprender las ideas principales del capítulo en el que se integra el material.
	<i>Prioridad media:</i> material complementario, ideas de interés o ejemplos y casos, cuyo estudio es recomendable para afianzar los conceptos bajo discusión.
	<i>Prioridad menor:</i> curiosidades, complementos o demostraciones que pueden ser dejadas para un estudio posterior, o para cuando sean referidas por otros capítulos más adelante. Obviamente, la recomendación de “dejarlo para luego” aplica especialmente a contenidos de “prioridad menor” y simultáneamente de “muchísima dificultad”.

Prerequisitos. Una revisión de los conceptos básicos de matrices, valores singulares, optimización y estadística aparece en el Apéndice A. Se recomienda su revisión previa para alumnos interesados en los contenidos de dificultad ***o superior. Las partes de modelado también tienen unos ciertos prerrequisitos de Física elemental, que deberían ser conocidos por los alumnos a partir de otras fuentes.

Recomendaciones de estudio

- Básicamente, un estudiante de *grado* debería visionar y comprender los contenidos etiquetados con dificultad *y **, y pasar por encima de bastantes de los contenidos de dificultad ***, dejándolos para una profundización posterior en una segunda pasada, quizás sólo en aquellos aspectos concretos que sean de su interés. Posiblemente, para estudiantes sin conocimientos previos existan otros materiales más adecuados, o/y deba formarse en contenidos de Física, Álgebra Lineal y Ecuaciones Diferenciales Ordinarias que aparecen en la troncalidad de los grados de ingeniería de la rama industrial.

- En el caso de un estudiante de *especialización de grado* o de *máster* con competencias ya adquiridas en fundamentos de Automática, los prerrequisitos del Apéndice A y los contenidos hasta dificultad ***serían los aconsejables inicialmente (posiblemente ya conocerá una parte significativa de ellos) y, una vez comprendidos, pasar a ****y superior.

En todos los casos, se recomienda revisar los contenidos más importantes de cada capítulo () y las ideas complementarias adicionales (), aunque si el lector ya es conocedor de parte de las ideas o desea simplemente un vistazo rápido, podría plantearse estudiar sólo los materiales marcados con . Los contenidos opcionales () tienen también interés, pero su estudio quedaría a discreción del lector, dependiendo del tiempo del que disponga, y de su interés o conocimientos previos en la materia tratada en cada capítulo.

- El **resumen y notas de texto** asociados a cada vídeo son **importantes**: para hacer a los vídeos relativamente “autocontenidos”, ni en el audio ni en los materiales (transparencias, Matlab) anejos se hace referencia a las relaciones con otros vídeos (dado que pueden cambiar en un futuro). Sin embargo, en las notas en este documento sí se relaciona con contenidos anteriores y posteriores, y se resumen/aclaran las ideas clave del vídeo. Por ello, *se recomienda su lectura “por encima” antes y “en profundidad” después del visionado*, para afianzar conceptos e integrar dicho vídeo en el esquema global del curso que se esté siguiendo.

Software de ejemplos

La mayoría de los ejemplos con código utilizan Matlab [<https://es.mathworks.com/products/matlab.html>], con *Control Systems Toolbox*, *Statistics and Machine Learning Toolbox*, *System Identification Toolbox* y *Robust Control Toolbox*. También el software libre GNU OCTAVE, ver [<https://www.gnu.org/software/octave/>], con el *control package* permite ejecutar bastantes de los ejemplos (de los que requieren control systems, robust control), con cambios mínimos de sintaxis respecto a Matlab. Algunos de los ejemplos de control robusto utilizan dicho software, para ilustrar dicha compatibilidad. No obstante, la compatibilidad Matlab↔GNU octave no es total, y hay muchos ejemplos Matlab en estos materiales que **no** pueden ser directamente ejecutados en Octave (en particular los ficheros interactivos “live scripts”, extensión **.mlx**). Enlaces a material introductorio sobre Matlab aparecen en la página 15. Se recomienda Matlab **R2019a** o superior.

Versiones de código: Durante los vídeos, en muchos casos se modifican fragmentos de código para ilustrar diferentes situaciones de simulación u opciones de diseño. Además, se puede haber incorporado variaciones menores o comentarios al código fruto de discusiones, clases, o actualizaciones de software, posteriores a la grabación. Por ello, el código Matlab disponible para descargarse no tiene por qué ser coincidente con el visualizado. Aparte, aquellos fragmentos de código donde entren en juego señales aleatorias (**rand**, **randn**, ...) pueden producir diferentes resultados, obviamente.

Renuncia de responsabilidad/mantenimiento: La publicación de ejemplos y ficheros de código no implica que estos vayan a ser activamente mantenidos y actualizados en el futuro. Funcionaron en su momento, con alguna versión/configuración concreta de Matlab, Octave, toolboxes, librerías y sistema operativo. No

obstante, esto no garantiza su funcionamiento en otros computadores, ni existe compromiso en atender solicitudes de soporte, particularmente en versiones anteriores o posteriores a la utilizada en la grabación del vídeo, debido a los distintos cambios que se van introduciendo en cada nueva versión del software por parte de sus desarrolladores. El código, asimismo, se realizó con fines expositivos/ilustrativos, y no tiene por qué ser la mejor implementación desde el punto de vista de consumo de recursos computacionales (memoria, tiempo) o de precisión numérica del resultado final, ni tiene que por qué producir soluciones “listas para su implementación práctica” dado que faltan muchos pasos de “validación”, “test de confianza estadística” o “simulación exhaustiva ante no-linealidades” y estudio de seguridad (comportamiento ante fallos en los sistemas controlados).

II.1.1 Otros materiales/cursos online alternativos

Herramientas Software

► Se supone como pre-requisito básico para la comprensión de estos materiales la familiaridad con Matlab. El profesor Juan Manuel Herrero, colega de mi Departamento, ha publicado unos breves vídeos de tutorial introductorio: [1.-Interfaz básica], [2.-Matrices], [3.-Scripts y bucles] [4.-Funciones], [5.-Gráficos].

► En <https://controlautomaticoeducacion.com/matlab/> también puede revisarse un curso básico de Matlab.

Cursos de nivel básico

► El profesor Pedro Albertos, catedrático emérito de la Universitat Politècnica de Valencia, es autor de un MOOC sobre “Dynamics and control” (<https://www.edx.org/course/dynamics-control-upvalenciav-dc201x-2>) el cual es probablemente de más interés para estudiantes de grado sin un background previo en el área.

► El profesor Antonio Barrientos (Universidad Politécnica de Madrid) también tiene un curso sobre dinámica de sistemas a nivel *grado*, con objetivos similares al anterior, con más de 140 vídeos didácticos, en <https://www.youtube.com/playlist?list=PL0zBmXWf2o8vXA9p-ZlHkZFEV-EgK8yLF>.

Cursos avanzados

Otros cursos avanzados de interés, mencionados posteriormente, son las siguientes colecciones de vídeos:

► **Introducción a la solución numérica de ecuaciones diferenciales con Octave**, UPV,
<https://www.youtube.com/playlist?list=PL6kQim6ljTJsKHdZUnYB4zm0wy4056WKE>

► **Control Predictivo basado en modelo (MPC)**:
<https://controlautomaticoeducacion.com/control-predictivo/>

► **Control Óptimo, Optimización Dinámica en sistemas continuos no lineales**:
<https://www.youtube.com/watch?v=Tk03YXaemFo&list=PLXmyaEzxlCQp0wVTxfiFuLdUqR7K0GiP>

Parte I

Modelado y Análisis del Comportamiento Dinámico de Sistemas

Capítulo 1

Modelado de sistemas físicos

Motivación: automática y control de procesos

[1: automintro] Automática, el control de los procesos

UNED et al.

*  17:07
[PLAY ►]

Este vídeo presenta, de forma divulgativa, los problemas que el área de conocimiento de *Ingeniería de Sistemas y Automática* aborda tanto de forma tecnológica (aplicaciones) como científica (teoría), con ejemplos en aplicaciones prácticas de los mismos.

Otro par de vídeos de la colección que también presentan las generalidades sobre qué es el “control” y qué tecnologías se utilizan son los vídeos [299] y [300], cuya visualización se aconseja.

[2: introAutI] Fundamentos de la Automatización Industrial

Instrumentacionycontrol.net

*  05:00
[PLAY ►]

Este vídeo introduce los conceptos básicos y elementos de la automatización en un proceso: sensor, setpoint, variable controlada, variable manipulada, transductor...

1.1 Señales y sistemas

[3: mod1ssa] Sistemas y Señales: conceptos básicos

Materiales: [SenyalesySistemasDefIntro.pdf]

*  14:41
[YouTube ►]

Este vídeo describe los conceptos de:

- magnitud física (información), señal (información función del tiempo), parámetro constante, parámetro variante en el tiempo
- Sistema (elemento que relaciona señales),
- modelos matemáticos de los sistemas, analogías

La clasificación de las señales y sistemas basándose en propiedades de los mismos y el rol de las señales en cada sistema se aborda en el vídeo [4], continuación de éste.

Un ejemplo motivador de los conceptos de señales y sistemas sobre una simulación interactiva de un tanque de almacenamiento de líquido se discute en los vídeos: [5](señales), [11] (utilidad de los modelos matemáticos).

[4: mod1ssb] Sistemas y Señales (clasificación)**Materiales:** [SenyalesySistemasDefIntro.pdf]**  10:33
[YouTube ▶]

Este vídeo, basándose en los conceptos de señales y sistemas definidos en el vídeo [3] (resumidos durante el primer minuto y medio aquí), discute, en términos muy generales, sobre:

- Clasificación de sistemas: estáticos/dinámicos, recurrentes, causales, realizables, ...
- Clasificación de señales según su rol en un sistema:
 - entradas: que varían arbitrariamente (bien manipuladas por usuario, bien como “perturbación” sobre cuya variación no tenemos control),
 - salidas de interés para una aplicación concreta,
 - Variables internas, variables de estado (energía).

Una ilustración conceptual ampliada sobre un depósito de estas ideas se aborda en el vídeo [5].

[5: tanksist] Señales, sistemas, dinámica: ejemplo cualitativo depósito de líquido*  11:30
[YouTube ▶]


Este vídeo, complemento de los vídeos [3] y [4], presenta el concepto de *señal*, *sistema*, *entrada* y *salida*, y de *sistema dinámico* analizando el comportamiento de un depósito de líquido (se simula en Matlab para entender por sentido común su comportamiento, para ilustrar más claramente las características).

Aunque no es necesario en este momento (estamos en temas muy introductorios), si te interesa entender el código de la animación del depósito, visualiza el vídeo [40] y descarga los ficheros Matlab asociados.

[6: reppyblk] Representaciones interna/externa/bloques de un sistema
controltheoryorg (YouTube)*  10:06
[PLAY ▶]

Este vídeo describe el concepto de representación “externa” de un sistema (elemento que procesa señales de entradas y salidas), representación “interna” (detalle de cómo se procesan dichas señales, apareciendo otras señales intermedias) y la interconexión de sistemas en representaciones “por bloques”.

- ▶ *Systems and signals: examples:* [\[Pedro Albertos\]](#)
[Enlace]
- ▶ *Systems and signals: representations:* [\[Pedro Albertos\]](#)
[Enlace]
- ▶ *Systems and signals: Formalism:* [\[Pedro Albertos\]](#)
[Enlace]
- ▶ *Systems and signals, Math review:* [\[Pedro Albertos\]](#)
[Enlace]

[7: sistB] Sistemas: definición, interconexión y propiedades.
Pablo A. Bernabeu Soler (UPV)*  08:18
[PLAY ▶]

Este vídeo presenta los sistemas como transformaciones de señales de entrada a salidas. Presenta sistemas continuos y discretos. Pone como ejemplo continuo un modelo de circuito RC y su modelado en ecuaciones diferenciales. Como ejemplo de un sistema discreto pone una cuenta del banco al 1% de interés mensual nominal. Plantea los problemas de análisis de “propiedades” de sistemas y de la “interconexión” de los mismos (serie, paralelo, realimentación), y del concepto de “entrada” y “salida”.

[8: sistMemB]

La propiedad de memoria en sistemas

Pablo A. Bernabeu Soler (UPV)

*  06:16

[PLAY ►]

Este vídeo describe la propiedad de “memoria” en sistemas: los sistemas sin memoria (estáticos) son aquéllos cuya respuesta no depende del “pasado” mientras que en los sistemas con memoria (dinámicos), la salida “presente” depende de valores de señales en instantes “pasados”.

1.1.1 Concepto de “estado” en sistemas físicos

[9: estado1]

Interpretación física del concepto de “estado” de un sistema dinámico

**  10:48**Materiales:** [ConceptoEstadoFis.pdf]

[YouTube ►]

En este vídeo se enuncian las hipótesis de “causalidad” de los sistemas físicos (el presente no depende de eventos futuros) y la hipótesis de que “las leyes físicas no tienen memoria” (el pasado influye en el futuro solamente a través del presente).

La interpretación física de la segunda hipótesis es que debe existir una energía o información almacenada “ahora” que resuma toda la historia pasada y la capacidad de realizar cambios en el futuro. El conjunto de todas esas variables se denomina vector de estado.

Discutir cómo plasmar esa hipótesis en los modelos matemáticos de los sistemas físicos será el objetivo del vídeo [10].

[10: estado2]

Ecuaciones de estado y de salida en modelos matemáticos realizables (Markov)

***  12:50**Materiales:** [ConceptoEstadoMat.pdf]

[YouTube ►]

Este vídeo es continuación del vídeo [9], donde se discutió las hipótesis sobre las leyes de la Física que dan lugar a la necesidad de que exista un cierto vector de estado que almacene toda la energía o información de la que dependerá la evolución futura de un sistema.

En los modelos matemáticos, implica la existencia de un conjunto de “variables de estado” $x(t)$ que verifique una *ecuación de estado*: $x(t+h) = f(x(t))$, o si el sistema no está aislado $x(t+h) = f(x(t), \mathcal{U})$ siendo \mathcal{U} la trayectoria de las entradas (potencias/información compartidas con otros sistemas) posteriores a t . Con $h \rightarrow 0$, se llega a la representación en tiempo continuo $\frac{dx}{dt} = f(x(t), u(t))$.

También se discute el caso de modelos inciertos (procesos de Markov, series temporales), donde las leyes físicas inciertas son “sin memoria” si se verifica que:

$$\begin{aligned} \text{Prob}(\mathbf{x}(t+h) = \xi \mid \mathbf{x}(t) = \xi_t, \mathbf{x}(t_1) = \xi_{t_1}, \dots, \mathbf{x}(t_m) = \xi_{t_m}) \\ = \text{Prob}(\mathbf{x}(t+h) = \xi \mid \mathbf{x}(t) = \xi_t) \end{aligned}$$

En algunos casos x sólo puede ser observada de forma parcial. Ello requiere introducir lo que se denominan ecuaciones de salida $y(t) = h(x(t), u(t))$. En general, las variables y , función de un subconjunto de x no verifican la hipótesis de Markov; se pone un ejemplo de lecturas de posición de un avión para ilustrar dicha situación.

Los modelos con ecuación de estado + ecuación de salida se denominan modelos en representación interna, modelos de Markov parcialmente osbservables, modelos ocultos de Markov, ... en distintas fuentes de la literatura.

La última parte del vídeo define la representación interna normalizada lineal $\dot{x} = Ax + Bu$, $y = Cx + Du$ que resulta de obligar a f y h en las expresiones genéricas citadas anteriormente a ser funciones lineales.

Este vídeo trata los modelos matemáticos de forma puramente “genérica” como consecuencia de la hipótesis de Markov (leyes físicas sin memoria), discutiendo únicamente la “forma” que deberían tener los modelos. Ejemplos de modelos físicos concretos y su representación interna aparecen en los vídeos [15], [16], [17], [23], ...

Los concepto de estado/Markov están relacionados con el concepto de *independencia condicional*; si se tiene una mínima base de estadística, se aconseja visualizar el vídeo [962] para relacionarlo con lo aquí presentado.

Los conceptos de causalidad y realizabilidad, para el caso lineal, son analizados en mayor profundidad en el vídeo [267].

1.2 Modelos físicos en forma de ecuaciones diferenciales

[11: modelsintro]
Modelos matemáticos y computacionales (digital twin) de sistemas físicos: motivación, utilidad

** 13:25

[\[YouTube ►\]](#)

* [LINK TO ENGLISH VERSION](#)

Este vídeo introductorio presenta los conceptos de modelos de sistemas físicos en tiempo continuo y sus principales aplicaciones, como motivación al estudio posterior en detalle de los mismos (en otros materiales y vídeos).

Aparte de los modelos físicos reducidos (prototipos a escala), la ingeniería usa modelos matemáticos en forma de ecuaciones diferenciales, y también se usan modelos computacionales (gemelo digital, digital twin, en el léxico del marketing actual. rimbombante) como herramientas para conseguir que las cosas funcionen conforme se especifique.

En concreto, los modelos se pueden usar en la fase de proyecto para diseñar el sistema y realizar simulaciones por computador de su comportamiento antes de su construcción. Una vez construido el sistema, un modelo puede ser ejecutado para comprobar el efecto de ciertas maniobras, antes de realizarlas sobre el sistema físico real con coste de materiales, energía o riesgo para personas o bienes. También puede ser utilizado para simular las trayectorias de señales de las que no se dispone de sensores físicos (esta aplicación se denomina *observador* en teoría de control). Por último, puede ser usado para ajustar valores de parámetros constantes de modo que la simulación se parezca lo más posible al experimento, y si esos valores son anormales, entonces se ha detectado un fallo.

Por último, el vídeo describe las ideas básicas del diseño de sistemas de control basado en dichos modelos.

Obviamente, los detalles de todo esto constituyen el núcleo de los cursos de grado y máster de las asignaturas del área de Ingeniería de Sistemas, Automática, Robótica, etc. que este vídeo motiva y esboza.

Aunque no es necesario en este momento (estamos en temas introductorios), si te gusta la programación y te interesa entender el código de la animación del depósito, visualiza el vídeo [40] y descarga los ficheros Matlab asociados.

1.2.1 Conceptos básicos

[12: mod2sf]

Modelos matemáticos de sistemas físicos: leyes dinámicas, estáticas y balances

*



12:15

Materiales:

[StateSpaceAndDAE.pdf]

[YouTube ▶]

Este vídeo describe los tipos de ecuaciones que aparecen en los modelos matemáticos de sistemas físicos:


- Leyes físicas con dinámica: retardos, derivadas, integrales.
- Leyes físicas sin dinámica (muelles, resistencias, conducción de calor)
- Concepto subyacente de acumulación en leyes con dinámica (energía cinética, energía térmica, energía potencial electrostática, ...)

Como resultado de integrar todo ese tipo de leyes físicas en un sistema de ecuaciones (algebraicas+diferenciales) se obtienen lo que se denomina **modelos de primeros principios** de un sistema físico (más concretamente, se abordan los modelos de parám. concentrados, en contraposición a los modelos de parámetros distribuidos que contemplan ecuaciones en derivadas parciales; para un ejemplo de esos modelos, ver los vídeos [79] o [78]).

[13: ecdifdf]

Ecuaciones diferenciales: definiciones básicas

**



16:06

controltheoryorg (YouTube)


[PLAY ▶]

Este vídeo describe las ecuaciones diferenciales: dada una señal $y(t)$, expresión $F(t, y, y', y'', \dots) = 0$ es una ecuación diferencial ordinaria, también pone ejemplos de ecuaciones diferenciales en derivadas parciales. Define las ecuaciones diferenciales lineales (con coeficientes variantes en el tiempo), y las ecuaciones diferenciales de un sistema lineal invariante en el tiempo (LTI), y de un sistema LTI causal.

[14: mod2bp]

Modelos matemáticos en ecuaciones diferenciales: modelos bien planteados

**



09:10

Materiales:

[StateSpaceAndDAE.pdf]

[YouTube ▶]

Este vídeo describe de forma abstracta un modelo matemático: combinando leyes físicas con/sin dinámica y balances, se obtiene un sistema de s ecuaciones expresable como:


$$\mathcal{F}\left(\frac{dx}{dt}, x, y, u\right) = \mathbf{0} \in \mathbb{R}^s$$

que expresa la relación entre variables asociadas a acumulaciones de información o energía (x), otras variables internas y , y variables de entrada u . Un modelo estará bien planteado si, en un problema tecnológico concreto donde vaya a ser utilizado el número de ecuaciones (s) es igual al número de incógnitas (señales a calcular). Se ponen como ejemplo qué significan en este contexto problemas de *simulación* y de *control*.

[15: mod3t1]

Modelado: pasos para obtener un modelo de un sistema físico en forma de ecuaciones diferenciales

**



08:55

Materiales:

[StateSpaceAndDAE.pdf]

[YouTube ▶]

Este vídeo define una ecuación diferencial ordinaria (EDO) en forma normalizada $\frac{dx}{dt} = f(x, t)$. Pone, como breve ejemplo un sistema térmico sujeto a entrada senoidal.

La primera aplicación de estas ecuaciones diferenciales podría ser la simulación: a partir de x en un instante t , se puede aproximar $x(t+h) \approx x(t) + h \frac{dx}{dt}$ (integración numérica por método de Euler), pero hay otras metodologías más precisas (ver vídeo [32], por ejemplo), fuera de los objetivos de este primer vídeo.

Una vez definida qué es una EDO y para qué podría servir (simulación de sistemas), se plantean los pasos necesarios para resolver un problema de modelado de un sistema físico: identificar leyes físicas, dar nombres simbólicos a las señales (funciones del tiempo) intervinientes y establecer los parámetros constantes, escribir las ecuaciones que expresan matemáticamente las leyes físicas.

Se pone como ejemplo un sistema masa-muelle $F_M = \kappa(l - l_{nat})$, $\frac{dl}{dt} = v$, $M \frac{dv}{dt} = Mg - F_m$. Se hace un análisis detallado de cómo escribir los signos en los modelos: en sistemas dinámicos la dirección de las fuerzas cambia con el tiempo (el muelle oscila hacia arriba y hacia abajo) y, por tanto, el signo en las ecuaciones *no* es la orientación de la fuerza en un cierto sistema de referencia, el vídeo analiza sobre el ejemplo del muelle cómo establecer de modo correcto los signos en los modelos físicos.


La fase final de un problema de modelado consiste en revisar y comprobar que el modelo es completo: el número de ecuaciones algebraicas o diferenciales del modelo debe coincidir con el número de incógnitas (señales) a calcular.

Ejemplos

[16: ssmmaB]

Modelado en espacio de estados de sistema
masa-muelle-amortiguador

Aureliano Esquivel

**  09:05

[PLAY ►]

Este vídeo modela un sistema masa-muelle amortiguador y obtiene una representación en variables de estado (representación interna) del mismo considerando a la posición y a su derivada (la velocidad) como variables de estado.

Revisión de modelado de este sistema y la obtención de soluciones de la ecuación diferencial asociada se abordan también en el vídeo [103].


Un caso con varias masas y muelles se aborda en el vídeo [24].

Los modelos resultantes pueden ser utilizados para cálculo de respuesta temporal, bien simbólica o bien numérica, ver vídeos [32], [105].

[17: sslB]

Modelado en representación interna de circuito con 2
resistencias, bobina y condensador

Academática (YouTube)

**  15:36


[PLAY ►]

Este vídeo detalla el modelado de las ecuaciones de estado de un circuito con una fuente de alimentación, dos resistencias, una bobina y un condensador. Como sólo existen una bobina y un condensador como elementos dinámicos, el resultado es una representación interna de segundo orden lineal. Se plantea al final una ecuación de salida seleccionando la tensión en el condensador como dicha variable de salida.

[18: termet]

Modelado de un sistema térmico lineal de orden 4 en
representación interna

Materiales: [termicoModel.pdf]

**  10:59

[YouTube ►]

Este video modela un sistema térmico constituido por dos habitaciones, un pasillo entre ellas y una buhardilla. Con las capacidades caloríficas de cada una de las estancias y las constantes de transmisión de calor, se construye un modelo de cuarto orden en representación interna. El modelo es introductorio, muy simplificado: no se considera suelo, capacidad calorífica de materiales de construcción, radiación, viento, que el aire caliente sube hacia el techo, etc.

La simulación y análisis de propiedades de este sistema térmico se abordan en el vídeo [274].

[19: mpmm1]

Modelado motor-polea-muelle-masa (1): ecuaciones de la física
(elementales+balances)

Materiales: [MotorPoleaMuelleMasaModelo1.pdf]

**
29:59

[YouTube ▶]

Este vídeo plantea el modelado de un sistema donde un motor ejerce un par (entrada al sistema) que hace girar un eje con cierta inercia al extremo del cual existe una polea con una cuerda que sostiene a un muelle del que cuelga una masa en el otro extremo.

Se plantea cómo obtener de forma ‘ordenada’ un modelo matemático resoluble del sistema:

1. Identificar los elementos del sistema y sus ecuaciones básicas (¿qué hay?)
2. Escribir las ecuaciones (balances, resultantes) que describan la interconexión entre los distintos elementos
3. Clasificar los símbolos en señales de entrada, señales incógnita y parámetros constantes
4. Elegir sistema de referencia y asignar los signos en las ecuaciones de balance adecuados para que el movimiento tenga sentido físico correcto.

Se aplica esta metodología al sistema físico, primero sin rozamiento por simplicidad y luego se añade rozamiento al final del vídeo. En los vídeos siguientes de este caso de estudio se realizará la transformación del modelo a ecuaciones de estado y salida normalizadas (vídeo [20]), así como simulaciones en Matlab.

[20: mpmm2]

Modelado motor-polea-muelle-masa (2): forma normalizada,
ecuaciones de estado y de salida

Materiales: [MotorPoleaMuelleMasaModeloNorm.pdf]

**
12:55

[YouTube ▶]

Este vídeo presenta cómo obtener las ecuaciones de estado y salida en forma normalizada del modelo de un sistema mecánico motor-polea-muelle-masa cuyas ecuaciones ‘no normalizadas’ fueron obtenidas en el vídeo [19] previo al presente en el caso de estudio.


Las ecuaciones normalizadas deben tener la forma $dx/dt = f(x, u)$, $y = h(x, u)$ siendo x el vector de estado y u las entradas al sistema.

Ya no se necesita saber nada de Física en esta fase del modelado: se trata de meras manipulaciones de las ecuaciones de un modelo matemático que esté bien planteado. Las ecuaciones normalizadas se obtienen utilizando las ecuaciones iniciales no normalizadas ‘sin derivadas’, para eliminar las fuerzas y pares resultantes en las ecuaciones elementales ‘con derivadas’.

1.2.2 Modelos algebraico-diferenciales generales (índice 1)

[21: mod3t2] Transformación de ecuaciones algebraico-diferenciales a ecuación diferencial normalizada (repr. interna): caso índice 1

Materiales: [StateSpaceAndDAE.pdf]

***  14:37

[YouTube ▶]

Este material plantea el problema de transformar un modelo escrito en forma de ecuaciones algebraico-diferenciales $\mathcal{F}(\frac{dx}{dt}, x, y, t) = 0$ a ecuación diferencial ordinaria $\frac{dx}{dt} = f(x, t)$, $y = g(x, t)$.

Pone un primer ejemplo de un sistema donde sencillas operaciones algebraicas permiten dicha transformación: $\dot{T} - Q = 0$, $k \cdot (T_e - T) = 0$ se transforma a $\dot{T} = k \cdot (T_e(t) - T)$ con una mera sustitución.

Esta clase de sistemas se denominan de "índice 1"... existen otros sistemas de índice superior (discutidos en el vídeo [30]). La parte central del vídeo discute un modelo lineal genérico donde

$$\mathcal{F}(\frac{dx}{dt}, x, y, t) = M_d \frac{dx}{dt} + M_x x + M_y y + M_u u = 0$$

El sistema es de índice 1 si $[M_d \ M_y]$ es invertible, y se puede obtener una representación $\frac{dx}{dt} = Ax + Bu$, $y = Cx + Du$ que se denomina representación interna normalizada.

La parte final del vídeo aplica esos resultados a un modelo de dos masas unidas por un amortiguador $\dot{v}_1 = f_{12} + F$, $\dot{v}_2 = -f_{12}$, $f_{12} = 4(v_2 - v_1)$.


El despejar $\frac{dx}{dt}$ e y de un sistema de ecuaciones puede ser realizado mediante software, como por ejemplo el Symbolic Toolbox de Matlab. Se propone al lector un ejercicio de modelado de un sistema eléctrico, cuya solución aparece en el vídeo [23].

1.2.3 Más ejemplos

Nota: Se recuerda al lector que es imprescindible la familiaridad con Matlab para seguir estos materiales, y que vídeos introductorios han sido referidos en la página 15.

[22: modmix] Modelado de un tanque de mezclado y obtención de representación interna (Matlab)

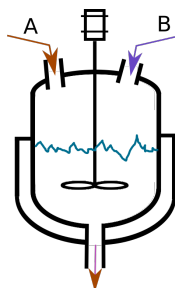
Materiales: [CÓD.: ModeladoTanqueMezclado.mlx] [PDF]

**  10:11

[YouTube ▶]

Nota: Se recomienda al alumno que, como ejercicio propuesto, intente resolver este problema de modelado antes de consultar la solución.

Este vídeo discute el modelado de un tanque de mezclado representado en la siguiente figura:



Donde dos caudales $q_{a,in}$ y $q_{b,in}$ de componentes A y B son mezclados en el tanque.


El objetivo es obtener un modelo de un sistema dinámico que permita calcular el nivel y concentraciones en función del tiempo. Una vez el modelo está escrito y se comprueba que está correctamente planteado, pasa a obtenerse una representación interna $\dot{x} = f(x, u)$, $y = h(x, u)$.

* El uso de Matlab es poco relevante: sólo se utiliza el comando **solve** para despejar determinadas variables (ponerlas “a la izquierda de un signo de igualdad”) a partir de unas ecuaciones escritas en forma y orden arbitrario. Pero, realmente, el modelo es sencillo y dichas operaciones deberían poder ser realizadas manualmente sin casi ninguna dificultad.

* El modelo no incluye dinámicas asociadas a la “medida” de la concentración o nivel. En aplicaciones reales, si la velocidad del agitador es poca y la viscosidad de los reactivos es alta, puede haber un gran retraso hasta que la medida de la concentración en la tubería de salida coincide con la concentración “media” del tanque aquí modelada. Del mismo modo, turbulencias, salpicaduras, ebullición, etc. podrían introducir mucho ruido en un sensor de nivel, que por tanto debería incluir un *filtro* adecuado. Estas consideraciones están fuera de los objetivos de este vídeo, pero se resaltan para indicar que el modelo aquí obtenido podría no ser lo suficientemente realista en muchas aplicaciones (un modelo más complejo del mismo tanque se aborda en los vídeo [537] y [538]).

[23: cir1] Modelado de un circuito electrico con 2 fuentes de alim
(variables de estado)

Materiales: [[CÓD.:](#) circ1mlb.mlx] [[PDF](#)]

**  12:54

[YouTube ▶]


Ejemplo de código para modelado de un sistema físico con Matlab:

- Este video describe el modelado de un circuito con 2 fuentes, 2 condensadores, 1 bobina y 3 resistencias.
- Se obtienen las ecuaciones algebraico-diferenciales y se despeja la forma normalizada (ecuaciones de estado y salida) lineal.
- Utiliza el Symbolic Toolbox de Matlab para despejar las ecuaciones.
- Utiliza la Control System Toolbox (**step**) para simular el transitorio ante escalón en las fuentes de tensión.

Sistema de 3 masas y 4 muelles

[24: moll3mod] Modelado de un sistema mecánico de 4 muelles y 3 masas
(representación interna, variables de estado)

Materiales: [Model3muellSpanish.pdf]

**  08:46

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo modela un sistema mecánico, de tres masas conectadas entre sí mediante muelles, y también conectadas tanto a una pared como a un extremo móvil cuya posición será la entrada al sistema. Habrá rozamiento con el suelo, pero no amortiguadores entre las masas. El sistema es similar al mostrado en el instante [00:50] del vídeo [100] que presenta experimentos reales de masa-muelle-amortiguador.

Aunque inicialmente se plantea en coordenadas absolutas, incluyendo longitud natural de muelles, rápidamente se pasa a coordenadas incrementales (respecto a la posición de equilibrio).

Se obtienen seis ecuaciones diferenciales de primer orden que constituyen la representación interna normalizada (las variables de estado son tres posiciones y tres velocidades).

Una revisión de ideas, fórmulas alternativas y expresión matricial normalizada $\dot{x} = Ax + Bu$ se aborda en el vídeo [25], continuación de éste. La simulación en Matlab se abordará en futuros vídeos.

[25: moll3mod2] Modelado de sistema mecánico de 4 muelles y 3 masas: recapitulación, expresión matricial $dx/dt=Ax+Bu$

Materiales: [3M4SModelSummary.pdf]

** 10:13

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo es continuación/complemento del vídeo [24], en el que se detalla cómo obtener cada ecuación de movimiento de un sistema de 3 masas y 4 muelles, y los signos de los balances de fuerzas.

En este vídeo se repasa el concepto de modelo en unidades absolutas (con longitudes naturales de muelles) y su punto de equilibrio (derivadas a cero). Luego se plantea el modelo en unidades incrementales y su expresión en forma normalizada $\dot{x} = Ax + Bu$, $y = Cx + Du$ calculando las matrices A y B ; respecto a la ecuación de salida, nótese que las matrices C y D dependen de cada aplicación, aquí se propone $C = I$, $D = 0$ como la solución más sencilla que 'saca todo' en la ecuación de salida, aunque posiblemente no la más frecuente en aplicaciones.

Para continuar con este caso de estudio, la simulación de este sistema de masas y muelles en Matlab se lleva a cabo en el vídeo [34], tanto con un integrador numérico genérico `ode45` como con el comando `lsim`, especializado en simular sistemas lineales.

Los autovalores de la matriz A son importantes en el estudio de las propiedades del sistema, en concreto de su respuesta libre, como el vídeo [171] analiza en detalle.

1.2.4 Caso de estudio sistema térmico (calentador de líquido)

[26: termrdmapEN] Tubular heater case study: ROADMAP

* 06:51

[YouTube ►]

This video presents the roadmap of a case study on the behaviour of a tubular heater, which will be discussed in a series of videos.

The videos will discuss from simplistic first-order models to partial difference equations, and intermediate options such as first-order time dynamics with non-uniform longitudinal temperature profiles or finite-element ones.

The first video of the case study with actual content is [??], and the links in its description will let you through the complete case study.

[27: term1e] Modelado dinámico de un tanque de calentamiento de líquido como un sistema de primer orden (mezclado perfecto)

Materiales: [[CÓD.:](#) HeatExch1elementModelado.mlx] [[PDF](#)]

*** 08:25

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo modela un tanque (agitado para que la temperatura sea homogénea en todo el volumen) donde una resistencia calefactora proporciona una potencia calorífica Q por el que pasa un caudal F de líquido de densidad ρ y calor específico C_e . La ecuación finalmente propuesta del modelo es $V\rho C_e \dot{T} = F\rho C_e(T_{in} - T) + Q - \kappa T$, y el vídeo la justifica en términos de balance de energía. Formalmente, como no se consideran variaciones de presión ni de volumen, los incrementos de "energía" y "entalpía" coinciden (no hay "trabajo" realizado).

La integración numérica (simulación) de este modelo con `ode45` en Matlab se aborda en el vídeo [33], y la linealización y la comparación de simulaciones lineal y no-lineal se abordan en el vídeo [94]; la representación de dicha linealización en función de transferencia se discute en el vídeo [183].

Obviamente, si el tanque de calentamiento no tuviera agitación perfecta o fuera, por ejemplo, un tubo largo, se necesitarían ecuaciones en derivadas parciales del tipo 1D en teoría de intercambiadores de calor (vídeo [79]) o mecánica de fluidos completa.

Aproximaciones, también de primer orden, a perfiles de temperatura no homogéneos si el elemento es largo (calentador tubular) se abordan en los vídeos [28] (perfil lineal de temperatura interior) y [29] (perfil exponencial a lo largo de la longitud inspirado en la solución de las EDP). Se recomienda la visualización de estos vídeos para entender que puede haber más de una forma de obtener una aproximación de primer orden a las ecuaciones de un calentador de fluido.

En este caso particular, para caudal constante, existe una solución exacta a la EDP (con retardo) que se obtiene en el vídeo [184]. La simulación comparada de las aproximaciones de primer orden y la solución EDP exacta se aborda en el vídeo [186].

[28: term1eP]

Modelado dinámico de tanque de calentamiento como sistema de primer orden (perfil temperatura lineal)

Materiales: [HeatExch1elementModeladoPadeyTF.pdf]

**** 12:54

[YouTube ►]

Este material describe el modelado de un sistema de calentamiento como un proceso de primer orden, de forma muy similar al vídeo [27]. No obstante, en dicho vídeo se discutía un escenario de agitación *perfecta* donde la temperatura de salida era exactamente igual a la temperatura “media” del interior del tanque de calentamiento.

El modelado del vídeo arriba referido es aquí resumido en los minutos iniciales de este vídeo, hasta el instante [03:00].

En caso de tanques “alargados”, “no agitados” y “relativamente bien aislados”, el perfil de temperatura podría aproximarse a una línea, de modo que la temperatura “media” (variable de estado de nuestro modelo) fuera $T = 0.5(T_{in} + T_{out})$ o, lo que es lo mismo, $T_{out} = 2T - T_{in}$. Ello hace necesario introducir modificaciones en la ecuación de estado y de salida de este sistema.

El resultado final resulta tener cierta similaridad con las aproximaciones de Padé del retardo de transporte que hay en las ecuaciones (en derivadas parciales, ver vídeo [79]) de una tubería alargada.

La linealización y expresión en función de transferencia ante cada una de las entradas F , T_{in} y Q se aborda en el vídeo [??].

Nota: El desarrollo en este vídeo (perfil lineal de temperatura) es un caso particular de los desarrollos con perfil exponencial de temperatura interna, inspirado en la solución estacionaria de las EDO, que se aborda en el vídeo [29], cuya visualización se aconseja. La simulación comparada con otros perfiles de temperatura y con la solución exacta de la ecuación en derivadas parciales de un calentador tubular se discute en el vídeo [186].

[29: term1exp]

Modelado dinámico de calentador tubular como sistema de primer orden (perfil temperatura exponencial)

Materiales: [EXPprofileHeatExch1elementModelado.pdf]

**** 17:58

[YouTube ►]

*[LINK TO ENGLISH VERSION](#)

Este vídeo modela un elemento intercambiador tubular donde una resistencia calefactora proporciona una potencia calorífica Q por el que pasa un caudal F de líquido de densidad ρ y calor específico C_e .

Se busca, intencionalmente, un modelo “simplificado” de primer orden, para evitar la complejidad de manejar modelos más ajustados a la realidad en ecuaciones en derivadas parciales (ver vídeo [79] para dichos modelos y [184] para tener idea de las funciones de transferencia resultantes).

No obstante, dado que las soluciones de las EDP tienen en régimen estacionario perfiles exponenciales respecto a la longitud, este vídeo propone usar dicho perfil exponencial en el balance de energía (potencia) siendo la variable de estado T la temperatura “media”. Con ello, la aproximación a la dinámica de un intercambiador tubular será más exacta (al menos en régimen permanente, equilibrio) que las obtenidas con suposiciones de temperatura “uniforme” (vídeo [27]) o de perfil “lineal” de temperatura interna (vídeo [28]).

Los modelos de primer orden desarrollados en los citados vídeos (de forma muy paralela a la argumentación aquí) serán un caso particular (límite para longitudes de tubo muy cortas o muy largas) del desarrollo aquí; por supuesto, la solución exacta no es de primer orden, sino EDP. El efecto de retardo de transporte es aproximado por términos de fase no mínima en la solución final.

La relación del enfoque con las famosas fórmulas LMTD (Logarithmic mean temperature difference, diferencia de temperatura media logarítmica) es también brevemente esbozada. Aquí somos algo más generales al permitir dinámica temporal y calentamiento interno.

La simulación de este modelo comparada con la solución exacta de la ecuación en derivadas parciales de un calentador tubular se aborda en el vídeo [186].

Este caso de estudio continua en la sección 4.6.4.

- El vídeo [lin1(27:01)] modela un sistema térmico con transmisión de calor por radiación (primer orden).
- Un sistema térmico de orden 2 (modelo de tubería) es abordado en la parte inicial del vídeo [tuboml(09:50)].
- Un sistema mecánico de orden 4 es modelado y linealizado en el vídeo [mod2mass(08:19)].

1.2.5 Modelos algebraico-diferenciales de índice superior

Transformación de ecuaciones algebraico-diferenciales a
ecuación diferencial normalizada (repr. interna): caso índice superior

⌂ 11:13

Materiales: [StateSpaceAndDAE.pdf]

[YouTube ▶]

Este material plantea el problema de transformar un modelo escrito en forma de ecuaciones algebraico-diferenciales $\mathcal{F}(\frac{dx}{dt}, x, y, t) = 0$ a ecuación diferencial ordinaria $\frac{dx}{dt} = f(x, t)$, $y = g(x, t)$.

En concreto, se estudian los sistemas de índice superior. Si tenemos un modelo lineal genérico donde

$$\mathcal{F}(\frac{dx}{dt}, x, y, t) = M d\frac{dx}{dt} + M_x x + M_y y + M_u u = 0$$

El sistema es de índice superior si $[M d \ M_y]$ tiene un rango inferior al número de ecuaciones. Ello requiere añadir “derivadas de ecuaciones” para poder despejar.

Como ejemplo, se aplica la idea a un modelo de dos masas unidas por un amortiguador $\dot{v}_1 = f_{12} + F$, $\dot{v}_2 = -f_{12}$, $v_1 = v_2$, esto es donde f_{12} sea la que haga falta para que se respete la *ligadura* mecánica que iguala las velocidades. El sistema es de índice superior, pero añadiendo $\dot{v}_1 = \dot{v}_2$ ya se puede obtener una representación interna.

Como ejemplo adicional, el sistema de engranajes y poleas modelado en el vídeo [170] es también, en la forma (arbitraria) en la que inicialmente se han escrito las ecuaciones, de índice superior, por lo que es necesario derivar alguna de ellas para obtener la representación interna. Se recomienda al lector el visionado del referido vídeo.

A partir del minuto [05:40] se analiza que las dos masas ligadas mecánicamente se comportan como una única masa, con lo que la complejidad del sistema (orden) podría reducirse.

Se comprueba que $\xi = v_1 - v_2$ se mantiene constante; esta combinación lineal de los estados que no cambia valga lo que valga la fuerza de entrada F se denomina *estado no controlable*.

En este vídeo se elimina el estado no controlable por “sentido común”, para obtener lo que en teoría de sistemas se denomina “realización mínima”. El análisis detallado de los conceptos de controlabilidad y realización mínima se aborda en los vídeos [279], [282].

En la parte final, se propone al lector un ejercicio de modelado de un sistema eléctrico, con dos condensadores en paralelo, para afianzar los conceptos discutidos en este vídeo.

[31: mod3t4]

Ecuaciones algebraico-diferenciales: consideraciones adicionales y conclusiones

07:40

Materiales: [StateSpaceAndDAE.pdf]

[YouTube ▶]

En este vídeo se presenta la representación normalizada de sistemas de ecuaciones algebraico-diferenciales lineales

$$E\dot{x} = Ax + Bu, \quad y = Cx + Du.$$

Si $E = I$, la representación interna es un caso particular, si E es invertible, entonces se puede transformar a la representación interna $\dot{x} = (E^{-1}A)x + (E^{-1}B)u$ –vídeo [21]–, y si E no es invertible, entonces ya se trata propiamente dicho de un sistema algebraico-diferencial de índice uno o superior.

A continuación, se analiza la transformada de Laplace de estas ecuaciones... dado que todo es “algebraico”, una vez transformado, entonces si un modelo está bien planteado, se “despeja” lo que se desee simular y ya está, $y(s) = (C(sE - A)^{-1}B + D) \cdot u(s)$.

Los sistemas mecánicos con ligaduras (que igualan posiciones), suelen ser modelados como ecuaciones algebraico-diferenciales de orden superior, que requieren añadir la igualdad de velocidades y aceleraciones para poder obtener una representación en variables de estado. Ese es el caso, por ejemplo, del sistema de engranajes del vídeo [170].

Por último, existe una metodología de modelado (ecuaciones de Euler-Lagrange) que evita variables intermedias (fuerzas de reacción, por ejemplo) en bastantes casos y que permite obtener también representaciones en forma de ecuaciones diferenciales ordinarias (no algebraicas) de una forma más directa. Esto está fuera de los objetivos de este vídeo.

La symbolic toolbox de Matlab tiene herramientas de manejo de ecuaciones algebraico-diferenciales no lineales, también fuera de los objetivos a este nivel. Existe software de modelado (Simscape, Modelica, ...) que también facilita el modelado y la simulación de sistemas complejos sin necesidad de conocer los detalles de las ecuaciones algebraico-diferenciales únicamente esbozados en estos materiales.

Otros ejemplos: Se remite al lector a los ejemplos adicionales que combinan modelado con cálculo de respuesta temporal, presentados en la Sección 4.6.

Nota: Los modelos en representación interna pueden obtenerse, además de mediante ecuaciones de primeros principios de la física, a partir de datos de sistemas multivariados: la identificación de modelos “caja negra” (desconociendo la física subyacente) en el caso *lineal* da lugar a lo que se conocen como métodos de *identificación subespacio*, discutidos en el Capítulo 23. *****

Videos de terceros relacionados:

► *Systems and signals: analogies:*

[Pedro Albertos]

[Enlace]

1.3 Autoevaluación

1. ¿Cuál es la diferencia fundamental entre un sistema estático y un sistema dinámico?
2. Proporciona ejemplos de ambos tipos de sistemas
3. Explica la diferencia entre una señal de entrada manipulada y una perturbación en el contexto de sistemas.
4. ¿Qué significa que un modelo matemático sea "de primeros principios"?
5. ¿Qué son las "variables de estado" en un sistema dinámico y cuál es su importancia?
6. ¿Qué es un modelo matemático y cuál es su propósito al representar un sistema físico?
7. ¿Qué es una ecuación diferencial ordinaria (EDO) y cómo se relaciona con la simulación de sistemas?
8. ¿En qué consiste la forma normalizada de un modelo dinámico (ecuación de estado y ecuación de salida)?
9. ¿Qué es la hipótesis de causalidad en sistemas físicos?
10. Explica la hipótesis de Markov y su significado en el contexto de sistemas dinámicos.
11. Describe un ejemplo de cómo la hipótesis de Markov se aplica en el contexto de un sistema informático.
12. ¿Qué son las variables de estado y por qué son importantes en la descripción de un sistema dinámico?
13. ¿Cómo se relaciona el concepto de "energía almacenada" con las variables de estado en sistemas físicos?
14. Explica la diferencia entre una ecuación de estado y una ecuación de salida en un modelo de espacio de estados.

Capítulo 2

Simulación de sistemas y casos de estudio adicionales

2.1 Simulación (integración numérica)


La simulación (integración numérica) es una disciplina en sí misma bastante amplia, dado que tiene amplias aplicaciones en casi todas las ramas de la ingeniería y de la física y, por tanto, su estudio y el diseño de código que la lleve a cabo de forma eficiente a gran escala es un problema de gran importancia hoy en día.

Se remite al lector a los libros de texto adecuados (cálculo numérico) si desea profundizar en el conocimiento de estas técnicas, dado que no tengo material elaborado/clasificado que estudie en profundidad dichos temas. Un curso completo de resolución numérica de ecuaciones diferenciales (35 vídeos), con profesores de la Universitat Politècnica de València, aparece en:

Introducción a la solución numérica de ecuaciones diferenciales con Octave,
UPV,

<https://www.youtube.com/playlist?list=PL6kQim61jTJsKHdZUnYB4zm0wy4056WKE>

El nivel de detalle de dicho curso es, no obstante, mucho más profundo del que se necesita para aplicarlo a nivel de mero “usuario” de las rutinas de Matlab, como se detalla a continuación. El curso estaría indicado para quien quisiera saber en detalle el funcionamiento interno de los comandos utilizados en los siguientes vídeos.

[32: sim1]	Simulación Matlab (Euler, ode45) de un sistema masa-muelle-amortiguador	**  14:42
	Materiales: [CÓD.: sim1.zip] [PDF]	[YouTube ►]

Este video presenta ejemplos de código Matlab de simulación (integración numérica).

NOTA: aunque el vídeo está grabado presentando el código en formato M-file (extensión *.m), versiones de Matlab recientes (a partir de 2016) incorporan la posibilidad de ficheros *Live Script* (extensión *.mlx) que permiten utilizar un único fichero (la función con el modelo al final del mlx) e incorporar texto y fórmulas para explicar el código. En el fichero ZIP de materiales está la versión original con la que se grabó el vídeo y la nueva versión MLX que, posiblemente, sea más recomendable en la actualidad (recomendación meramente “estética” porque los cálculos realizados son exactamente los mismos).

- El vídeo describe las opciones más importantes de `ode45` (Matlab).
- Describe cómo generar una función de Matlab con el modelo (el argumento `odefun` para `ode45`) para simular un muelle lineal sujeto a una fuerza externa (función del tiempo) de entrada.

- Integra (simula) esa función mediante el método de Euler, programado en un bucle, y lo compara con el resultado de `ode45` para diferentes valores de paso de simulación h . Otro ejemplo de simulación por Euler de un depósito de primer orden aparece en el video [40] (bueno, se trata de una simulación interactiva por lo que también discute código de dibujado/animación y respuesta a teclado que no es estrictamente de integración numérica).

Como ejemplos adicionales, el video [34] simula un sistema de varias masas y muelles; el video [64] simula con `ode45` el movimiento de una bola ensartada en un alambre.

Los sistemas pueden ser más complejos e integrar, por ejemplo, “controladores”; un bucle cerrado de control se simula con `ode45` en el video [355].

A veces, los resultados de simulación pueden ser entendidos mejor mediante animaciones: el video [35] anima un sistema masa-muelle con rozamiento, y de hecho, la simulación de ecuaciones diferenciales es el elemento fundamental de los “motores de física” de los videojuegos. Código para otras animaciones de sistemas mecánicos puede verse en el video [35].

La comparación entre simulación numérica y resolución simbólica de ecuaciones diferenciales ordinarias lineales se discute en el video [105].

La simulación está en la base de lo que se denomina “identificación experimental”: el video [749] ajusta la masa y constantes de amortiguador y muelle basándose en datos de laboratorio, su visualización puede ser aconsejable.

[33: term1sim]

Simulación (ode45) de un tanque de calentamiento de primer orden

Materiales: [[Cód.:](#) HeatExch1elementSIM.mlx] [[PDF](#)]

** 10:43

[[YouTube](#) ►]

Este video modela un tanque (agitado para que la temperatura sea homogénea en todo el volumen) donde una resistencia calefactora proporciona una potencia calorífica Q por el que pasa un caudal F de líquido como un modelo de primer orden. El modelado se describe rápidamente en los primeros dos minutos y medio; si se desea una descripción más detallada de la obtención del modelo, se puede visualizar el video [27].

A continuación, se presenta código donde se transforma el modelo de la Symbolic toolbox a una función ordinaria (numérica) de Matlab, y se describe cómo crear una función “envoltorio” (wrapper) de modo que el modelo sea llamado con los argumentos necesarios para la simulación con `ode45`.

Dando valores a las entradas Q , F y T_{in} como un tren de ondas cuadradas (pueden ser funciones del tiempo arbitrarias), se simula con `ode45` y se representa gráficamente la solución.

La parte final del video comprueba que la solución obtenida tiene “sentido físico” examinando los cambios de temperatura de salida ante cambios de caudal y temperatura de entrada. También se comprueba que los puntos de equilibrio (calculados haciendo $\frac{dT}{dt} = 0$) del sistema no lineal ante varios valores de las entradas coinciden con las simulaciones numéricas.

La linealización de este modelo de tanque calefactor y la comparación de simulaciones lineal y no-lineal se abordan en el video [94]. En este caso particular, para caudal constante, existe una solución exacta a la EDP (con retardo) que se obtiene en el video [184]. La simulación comparada de las aproximaciones de primer orden y la solución EDP exacta se aborda en el video [186].

[34: moll3sim1]

Sistema mecánico de 4 muelles y 3 masas: simulación ode45 vs lsim

Materiales: [[Cód.:](#) simul3M4Sstandalone.m] [[PDF](#)]

** 08:49

[[YouTube](#) ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo continua el caso de estudio de un sistema de 3 masas y 4 muelles cuyo modelado se abordó en los vídeos [24] y [25]. Aquí, se introducen las ecuaciones $\dot{x} = Ax + Bu$ en forma matricial, simplemente introduciendo las matrices constantes A y B copiándolas de las transparencias del modelado teórico. La ecuación de salida $y = Cx + Du$ se implementa de modo que sólo las tres posiciones formen parte del vector y .

Una vez introducido el modelo, se realiza una simulación ante unas ciertas condiciones iniciales y un perfil de entrada senoidal (elegidos arbitrariamente). Para comparar los resultados y el tiempo de cómputo, se realiza la integración con `ode45` (simulador genérico, que también podría simular sistemas no lineales) y con `lsim` (simulador especializado en sistemas lineales). El segundo de los casos es, al menos en este ejemplo, cuatro o cinco veces más rápido en ejecutar; ambas simulaciones son básicamente coincidentes, aparte de tolerancias numéricas en el quinto o sexto decimal.

La animación para visualizar mejor qué está ocurriendo en este sistema se discute en el vídeo [35]. Un estudio más formal de la respuesta libre de este sistema se discute en el vídeo [171].

[35: moll3ani]

Sistema mecánico de 4 muelles y 3 masas: ejemplos animación
respuesta libre y forzada

Materiales: [[CÓD.:](#) simul3M4Sstandalone.m] [[PDF](#)]

**
☰
11:11

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo presenta el código que realiza una animación de las vibraciones de un sistema de tres masas y cuatro muelles, cuyo modelo fue discutido en el vídeo [25], y la simulación con `ode45` o `lsim` fue abordada en el vídeo [34].

Depende de tu experiencia previa con Matlab, es posible que desees visualizar previamente el vídeo [102] que anima una única masa, para familiarizarte con una versión algo más simple del código.

Se simulan tres modos de vibración en frecuencias de resonancia (la obtención de dichas frecuencias queda fuera de los objetivos de este vídeo), así como una simulación de respuesta libre (cond. iniciales no nulas, entrada nula). Se hace animación de las vibraciones tanto suponiéndolas longitudinales como transversales.

Caso de estudio tanque de mezclado:

[36: odemix]

Simulación de un tanque de mezclado con Matlab (integración
numérica ode45)

Materiales: [[CÓD.:](#) ModelSimulacionTanqueMezclado.mlx] [[PDF](#)]

**
👤
10:18

[YouTube ▶]

En este vídeo se simula la respuesta temporal del tanque de mezclado cuyo modelo no lineal se desarrolló en el vídeo [22]. Para que sea más “autocontenido”, los dos primeros minutos resumen rápidamente esos pasos del modelado; si se ha visionado anteriormente el vídeo [22], puede comenzarse éste en el minuto [02:40].

En la ecuación de salida, en vez de incorporar todas las señales posibles, como se hizo en el vídeo antes referido, ahora sólo se desea obtener las trayectorias (respuesta temporal) del nivel del tanque h y de la concentración de componente B , x_B . Por ello se selecciona sólo un subconjunto de los resultados de `solve`.

La simulación pasa primero por convertir el modelo de la Symbolic Toolbox en una función ordinaria (numérica) de Matlab, con el comando `matlabFunction`. Ello se realiza tanto con las ecuaciones de estado como de salida.

Un segundo paso genera una función del tiempo interpolando entre los puntos (arbitrarios) de los perfiles de entrada $q_{a,in}$ y $q_{b,in}$, con el comando `interp1`. Con ello ya se tiene todo preparado para ejecutar `ode45` y obtener las trayectorias del estado (v_a, v_b) y con dichas


trayectorias sustituirlas (ya no hace falta más integración numérica) en las ecuaciones de salida.

Los sistemas que `ode45` puede simular podrían ser más complejos e integrar, por ejemplo, “controladores”; un bucle cerrado de control se simula con `ode45` en el vídeo [355]. Asimismo, en modelados físicos complejos de elevado orden o constantes de tiempo muy diferentes (stiff), es posible que otros integradores numéricos diferentes a `ode45` sean más eficientes computacionalmente (el vídeo [39] compara `ode45` con `ode15s`).

Caso de estudio motor-polea-masa-muelle:

[37: mpmm3]

Modelado motor-polea-muelle-masa (3): forma normalizada con Matlab (Symbolic Toolbox)


15:45


Materiales:
[[CÓD.:](#) MotorPoleaMuelleMasa24B.mlx]
[[PDF](#)]

[\[YouTube ▶\]](#)

Este vídeo presenta el cómo introducir las ecuaciones de la física derivadas en el vídeo [19] como expresiones simbólicas en Matlab, y luego realizar las operaciones adecuadas con el comando `solve` de la Symbolic Math Toolbox, para obtener la forma normalizada de ecuaciones de estado y de salida, idéntica a la obtenida en el vídeo [20] sin usar computador.

[38: mpmm4]

Modelado motor-polea-muelle-masa (4): simulación ODE45, análisis equilibrio


22:56

Materiales:
[[CÓD.:](#) MotorPoleaMuelleMasa24B.mlx]
[[PDF](#)]

[\[YouTube ▶\]](#)

Este vídeo es continuación del caso de estudio de un sistema motor-polea-muelle-masa que se está detallando en los vídeos anteriores [19] (modelado), [20] (representación interna normalizada), y [37] (manipulación del modelo en Matlab).

Aquí se analiza el equilibrio del sistema (derivadas de ecuación de estado igual a cero), y se simulan con `ode45` diferentes maniobras de respuesta libre (entrada equilibrio, estados iniciales separados de posición de equilibrio), respuesta forzada (entrada diferente a la calculada para mantener un punto de equilibrio dado) tanto desde condiciones iniciales de equilibrio como fuera de él, que sería el caso más general.


La parte final del vídeo pone como ejemplo que una vez se ha simulado la ecuación de estado, con dicha solución pueden calcularse los resultados de aplicarle la ecuación de salida para, por ejemplo, graficar la tensión de la cuerda en función del tiempo y comprobar que no se hace negativa, lo cual invalidaría el modelo.

Realmente el modelo es lineal (excepto unas constantes, formalmente es “afín”) de modo que es muy sencillo obtener una forma normalizada matricial $\frac{dx}{dt} = Ax + Bu$, $y = Cx + Du$ y usar la Control Systems Toolbox para simular y analizar el mecanismo; todo esto se discute en detalle en el vídeo [??], continuación del caso de estudio.

También se realiza simulación, por ejemplo, en los vídeos [lin1(27:01)] y [linmix(10:55)].

[39: ode45vs15s]

Integración numérica: comparación ode45 versus ode15s en casos rígidos / no rígidos (stiff/ non-stiff), ejemplo Matlab

**

08:14

Materiales:
[[CÓD.:](#) Stiffoder45vs15sTST.mlx]
[[PDF](#)]

[\[YouTube ▶\]](#)

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute la integración numérica de ecuaciones diferenciales ordinarias (en representación en variables de estado, sistemas de ecuaciones diferenciales de primer orden), comparando las prestaciones de `ode45` y `ode15s` en casos donde las constantes de

tiempo son todas similares (caso no "stiff", no "rígido"; en ese caso `ode45` es el que ejecuta más rápido) y en casos "rígidos" (stiff) con constantes de tiempo muy diferentes; en ese segundo conjunto de casos, es `ode15s` el que ejecuta mucho más rápido.

En el vídeo [188] también se observa, en un modelo de elementos finitos de un intercambiador de calor con retardo de transporte, que `ode15s` resulta más rápido que `ode45` pero, como se ha visto aquí, cada caso es diferente y debe ser estudiado individualmente para determinar el mejor compromiso entre exactitud y la rapidez de ejecución de la simulación.

La simulación de modelos es la base de la identificación experimental de parámetros físicos; podría ser de interés que visualizaras el vídeo [identga(19:49)] en este momento.

Simulación interactiva. Resulta de interés realizar simulaciones que interaccionen con usuarios mediante teclado, ratón o animaciones... pensad en la *física* de un videojuego. Aquí va un ejemplo (muy elemental comparado con los videojuegos de hoy en día, obviamente):

[40: interactank]

Simulación interactiva de un tanque de líquido: detalle código

☰
14:58

clase Matlab
Materiales:
[Cód.: TankAnimInteractive.zip]
[PDF]
[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo detalla el código Matlab para la animación interactiva (respondiendo al teclado) del llenado o vaciado de un depósito de líquido... es una aproximación inicial a lo que sería el subsistema de la *física* de un entorno de desarrollo de videojuegos, salvando las (enormes) distancias. Esta animación se utilizó en vídeos introductorios de motivación hacia la ingeniería de sistemas (vídeos [5] y [11]) previamente publicados.

Aquí se detalla la clase (`classdef`) que gestiona los datos de la animación y que tiene los métodos siguientes:

- *constructor*, para inicializar los datos;
- dibujador de "frames" para animación, 20 veces por segundo;
- simulación de la ecuación diferencial $\frac{dh}{dt} = \frac{1}{S}(u - \kappa\sqrt{h})$, aproximando por método de Euler. Si el período de muestro (de frame a frame, 1/20 s) fuera mayor, puede que se necesitara otro método de integración numérica, como por ejemplo `ode45`. Este método de la clase es el más importante conceptualmente desde un punto de vista de teoría de sistemas (integración numérica). El vídeo [32], por ejemplo, también aborda este tipo de simulación (pero no de forma "interactiva").
- Respuesta a evento 'tecla pulsada' para realizar distintas acciones en función de la tecla que el usuario haya activado. Las acciones posibles son subir/bajar caudal de entrada, o parar la simulación.

Asimismo, en la parte final del vídeo se detalla el código de un script que se encarga de crear la animación, simular durante un cierto tiempo y presentar gráficas de entrada y salida una vez finalizada la simulación.


Software de simulación gráfica. Introducir las ecuaciones del sistema y manipularlas para tener el código que necesita `ode45` resulta, en casos complejos, extremadamente tedioso. Por ello, hay muchos paquetes software de simulación que, mediante herramientas gráficas y bibliotecas de modelos físicos, intentan facilitar la tarea. Algunos paquetes están especializados en determinado tipo de sistemas, y otros son genéricos. Matlab incorpora `Simulink` y `Simscape`. Un esbozo de cómo usar el primero de ellos aparece a continuación.

[41: smkintr]
Introducción a Simulink
Juan M. Herrero Durá (UPV)

*
👤
03:37

[PLAY ▶]

Este vídeo introduce la funcionalidad básica de Simulink, indicando cómo abrirlo desde Matlab, el tipo de elementos (bloques) que pueden incorporarse en un modelo de simulación, y para ilustrar un ejemplo sencillo, genera una señal senoidal que multiplica por una constante y representa gráficamente el resultado. También exporta las variables de la simulación al espacio de trabajo de Matlab, si se quiere almacenar o procesar el resultado.


[42: simulmw] **Introducción a Simulink (Mathworks)** Javier Gazzarri, MathWorks ***  30:11
[PLAY ►]

Este vídeo modela un sistema masa-muelle-amortiguador con Simulink. Crea el modelo desde cero, usando bloques de ganancias, sumas e integradores, y lo encapsula en un “subsistema”. Los parámetros del sistema son introducidos en una “máscara” donde se diseña un cuadro de diálogo para introducirlos. La simulación se visualiza en un “scope”, pero también mediante el “logging” y el “simulation data inspector” de modo que pueden almacenarse distintas ejecuciones con distintos parámetros y visualizarlas.

En la segunda parte del vídeo, se introduce un bucle de control: se simula un controlador (PID) que ejerce una fuerza función de la diferencia de posición entre la posición medida y la posición deseada (referencia). Utiliza la sintonía automática integrada del sistema de control.

La última parte del vídeo implementa una máquina de estados (eventos discretos) trivial –podría ser estática–, mediante el módulo “StateFlow”, definiendo cuatro posibles estados del generador de referencias y los eventos de transición asociados.


Para simular sistemas no es necesario Simulink, dado que realmente es una interfaz de usuario a integradores numéricos (`ode45`, `ode15s`, ...). Se puede, por tanto, hacer simulaciones en un fichero Matlab, como se hace en los vídeos [32], [34], [39], etc.

[43: slkmix] **Modelo de tanque de mezclado: importación a Simulink** **  08:03
Materiales: [[CÓD.:](#) slkmix.zip] [[PDF](#)] [YouTube ►]

Este vídeo importa a Simulink las ecuaciones de estado y de salida obtenidas en el vídeo [22]; por hacerlo un poco más autocontenido, un breve resumen de dicho modelado se discute hasta el instante [01:44] del vídeo.

La simulación en Simulink es una alternativa a la del `ode45` discutida en el vídeo [36]. Simulink permite una interfaz de usuario gráfica incorporando generadores de ondas, osciloscopios, interacciones con otros sistemas, etc. que podría resultar más laborioso de preparar si se generara manualmente un modelo para `ode45` en casos complejos (realmente Simulink acaba traduciendo todo a una función para el integrador numérico, pero de forma casi transparente al usuario).

La clave de la importación a Simulink es el comando `matlabFunctionBlock`, que traduce una expresión del Symbolic Toolbox a un bloque `Matlab Function` de código Matlab dentro de un modelo o subsistema “.slx”. Dos llamadas a ese comando crean ecuaciones de estado y de salida; luego, se pone un integrador tras las ecuaciones de estado (para transformar las derivadas del estado a los estados propiamente dichos), especificando una condición inicial arbitraria. Esos estados son “realimentados” a la ecuación de estados (dado que las derivadas del estado dependen de ellos mismos), e insertados en la ecuación de salida. Los adecuados bloques `Demux`, `Inport` y `Outport` completan el modelo, que se guarda como un *subsistema* para ser utilizado como un componente de otros modelos más complejos.

[44: slkmix2] **Modelo de tanque de mezclado: simulación con Simulink de respuesta temporal** *  04:18
Materiales: [[CÓD.:](#) slkmix.zip] [[PDF](#)] [YouTube ►]

Este vídeo incorpora a un modelo de simulación el subsistema generado en el vídeo [43], mediante el bloque **Subsystem Reference**. Luego, genera formas de onda arbitrarias para las entradas q_a y q_b (mediante el bloque **Repeating Sequence (Stair)**), visualizando las entradas y salidas mediante los bloques **Scope** oportunos.

Nota: Simulink puede también hacer linealización (Capítulo 3), como se discute en el vídeo [97] sobre este mismo modelo.

Una vez introducidos conceptos adicionales (en concreto, función de transferencia), un ejemplo masa-muelle-amortiguador en Simulink se presenta en el vídeo [sssmk(11:46)], y de sistemas de depósitos en [smktf(06:27)].

2.2 Casos de estudio de modelado de sistemas mecánicos

Como el modelado de sistemas mecánicos y electromecánicos es de especial interés en ingeniería industrial en general y robótica en particular, esta sección discute algunos casos de estudio adicionales en detalle sobre ese tema. Algunos ejemplos utilizan ecuaciones de Euler-Lagrange, cuya teoría no está discutida en esta colección, pero es habitual en textos de Mecánica y Robótica.

2.2.1 Caso estudio: simulación interacción gravitatoria entre varios cuerpos

[45: nbdsimc]

Simulación N cuerpos bajo fuerza gravedad: código Matlab
 ode113, animación

******* **14:57**

Materiales: [[CÓD.:](#) NbodySim.m] [[PDF](#)]

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute el código de simulación en Matlab de la interacción gravitatoria de N cuerpos (movimiento plano 2D, por simplicidad). El simulador es **ode113**, y la ecuación de estado es que la derivada de la posición es la velocidad y la derivada de la velocidad es la aceleración de gravedad (fórmula de Newton). Cada cuerpo tiene posición 'x' y posición 'y' así como velocidades, por lo que el orden total es $4N$. Se explica todo el código de bucles que calculan las fuerzas para todos los pares de cuerpos y la aceleración resultante para cada cuerpo.

La parte final del vídeo discute brevemente el código para la animación del movimiento, simplemente con **plot**.

[46: nbdsim1EN]

N-body simulation of gravitational interaction examples:
 ellipses, chaos, escape velocity

****** **16:15**

Materiales: [[CÓD.:](#) NbodySim.m] [[PDF](#)]

[YouTube ▶]

This video presents examples simulating the motion of several bodies under gravitational interaction. The video uses the code detailed in video [??], briefly summarised in the first two minutes. Then, examples with 3, 4, and 5 bodies appear and emphasis is placed on the simulation of problems with two bodies, to illustrate elliptical orbits, barycenter motion (conservation of momentum), and escape velocity of a light body from the attraction of a (much) heavier body, via mechanical energy conservation arguments.

2.2.2 Mecanismo piñón-cremallera

[47: pinionm]

Modelo dinámico de sistema piñón-cremallera: ecuaciones de Newton, masa/momento inercia equivalente

Materiales: [RackPinion.pdf]

**
15:17

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute el modelado dinámico de un mecanismo piñón-cremallera de un grado de libertad.

Se presentan las ecuaciones de Newton (balances de fuerzas, incluyendo fuerza de reacción), y la ligadura cinemática entre desplazamientos lineal y angular (que hace proporcionales los incrementos de posición, velocidad y aceleración).

Con ese modelo completo, luego se presentan dos modelos simplificados equivalentes, visto desde el lado del engranaje (inercia equivalente) y visto desde el lado de la cremallera (masa equivalente). En efecto, como el modelo sólo tiene 1 grado de libertad, se puede reducir todo a una ecuación de orden 2 de masa o inercia equivalente por aceleración lineal o angular, respectivamente.

La parte final del vídeo calcula la fuerza de reacción entre piñón y cremallera, tanto en equilibrio (aceleración cero) como fuera de él.

Un enfoque equivalente basado en el formalismo Euler-Lagrange con restricciones aparece en el vídeo [??].

[48: pinionELEN]

Pinion-rack dynamical model: Euler-Lagrange equations with kinematic constraints

Materiales: [RackPinionEL.pdf]

17:13

[YouTube ▶]

This video discusses the Euler-Lagrange modeling of a 1 degree of freedom rack-and-pinion mechanism.

The mechanism is identical to the one seen in the video [??], where Newton's equations were used, with force and torque balances where reaction forces intervened in the contact.

The Euler-Lagrange equations do not involve reaction forces, but rather kinetic and potential energies (the latter is zero in this specific simple example).

The video uses the Euler-Lagrange formalism in two ways:

- Expressing the Lagrangian $L = T - V$ as a function of a single coordinate (1 degree of freedom model), obtaining a model with (inertia/equivalent mass) times acceleration equal to a certain balance of torques or generalized forces.
- Expressing the Lagrangian in terms of two generalized coordinates (2 degree of freedom model) and introducing the mechanical link between both, $f(q) = q_1 \rho - q_2 = 0$, through a Lagrange multiplier. Due to the way it is written, the multiplier ends up being equal to the reaction force, but that need not happen in a more complex problem setup, in general.

As a result of both developments, exactly the same models are obtained as these in the video referred to above (which used Newton's mechanics), obviously, given that they are equivalent.

2.2.3 2 masas y muelle, mediante Euler-Lagrange

Se plantearán ejemplos sencillos para afianzar ideas básicas (Hamilton es prescindible para estudiantes usuales de ingeniería):

[49: twomELH] [Modelado Euler-Lagrange / Hamilton de un sistema de dos masas unidas por un muelle](#)

Materiales: [[CÓD.:](#) DosMasasMuelleEnMedio.mlx] [[PDF](#)]

*** 11:18

[YouTube ▶]

Este video modela la dinámica de dos masas unidas por un muelle mediante la metodología Euler-Lagrange, a partir de las expresiones de energía cinética $Mv^2/2$ y energía potencial elástica $Kx^2/2$.

La parte final del vídeo (opcional) obtiene las ecuaciones del movimiento de Hamilton, coincidiendo (en este caso no forzado) el Hamiltoniano con la energía mecánica total.

Otro ejemplo sencillo de modelado Euler-Lagrange con restricciones de un sistema de 1GL aparece en el vídeo [??].

2.2.4 Movimiento lateral de un tiovivo

[50: tiovintr] [Modelado elemental de tiovivo/péndulo rotatorio: presentación del caso de estudios y la serie de vídeos](#)

* 04:12

[YouTube ▶]

Este vídeo presenta una introducción a qué se pretende conseguir con la serie de vídeos que abordan el caso de estudio de un tiovivo con un único pasajero colgando, por simplicidad. Se presenta la abstracción a péndulo giratorio de 2 grados de libertad, se define qué es la *dinámica lateral* y se incorpora una simulación/animación de las ecuaciones del movimiento para motivar el estudio de los sucesivos vídeos que desarrollan todo el caso de estudio.

[51: tiovs] [Modelado dinámica lateral tiovivo/péndulo rotatorio \(1\): planteamiento y estática \(Matlab\)](#)

Materiales: [[CÓD.:](#) tiovivo1GL.mlx] [[PDF](#)]

** 13:51

[YouTube ▶]


Este vídeo plantea el modelado de las oscilaciones laterales de un tiovivo. Realmente, sólo aborda un primer paso de planteamiento del problema, esquema de las fuerzas intervinientes y de escritura de un modelo de la “estática” (equilibrio horizontal y vertical de fuerzas: peso, centrífuga y tensión de cuerda). Se comenta que si se escribiese la dinámica directamente (vídeo [52]) entonces el modelo estático aquí desarrollado sería innecesario porque sería un caso particular del dinámico simplemente poniendo derivadas igual a cero, como se demuestra en el vídeo continuación de éste arriba enlazado.

El symbolic toolbox no encuentra con `solve` una forma “cerrada” de escribir el ángulo de equilibrio en función de la velocidad de rotación del tiovivo en su conjunto. Por ello, se usa `vpasolve` para obtener una solución numérica concreta, dando valores a masas y longitudes. Es posible que se pueda despejar usando alguna identidad trigonométrica, pero me ha dado pereza el intentarlo al ver que Matlab no ha podido. Nótese que la solución no es única (hay más de un punto de equilibrio, por ejemplo otro hacia arriba estilo “motocicleta en curva”, inestable), pero ha habido suerte y `vpasolve` ha dado un resultado que físicamente interpretamos que es correcto para nuestra aplicación.

La linealización aproximada de este tiovivo se aborda en el vídeo [89] .

[52: tiovd]

Modelado dinámica lateral tiovivo/péndulo rotatorio (2):
ecuaciones; simulación ode45 ecuación de estado
Materiales: [[CÓD.:](#) tiovivo1GL.mlx] [[PDF](#)]

***  17:58
[\[YouTube ▶\]](#)

Este vídeo es continuación del [51] donde se planteaba el problema del movimiento pendular lateral de un tiovivo y se calculaba el ángulo de desviación hacia afuera del columpio en función de la velocidad de giro del tiovivo, pero sólo en equilibrio (estática).


En este vídeo se plantean las ecuaciones de la dinámica completas, haciendo un balance de pares. Se comprueba que haciendo derivadas a cero se obtienen las ecuaciones estáticas del vídeo anterior con lo que, en cierto modo, dicho vídeo “sobra” si uno conoce bien el modelado dinámico de sistemas.

La linealización de este modelo dinámico se aborda en el vídeo [89], y también en el [90] linealizando la ecuación de estado no lineal normalizada (resultados coincidentes).

Una vez obtenido el modelo dinámico, se obtiene la ecuación de estado normalizada y se simula con `ode45`, comprobándose que sin rozamiento hay una oscilación mantenida alrededor del equilibrio estático antes calculado; se modifica el modelo incluyendo un rozamiento lineal proporcional a velocidad angular, con lo que se introduce un amortiguamiento que hace que las oscilaciones desaparezcan al cabo de un cierto tiempo, según se verifica en simulación.

[53: tiovELEN]

2DoF dynamics of a carousel (rotational pendulum):
Euler-Lagrange equations
Materiales: [[CÓD.:](#) tiovivo2GLEulerEN.mlx] [[PDF](#)]

****  19:50
[\[YouTube ▶\]](#)

This video discusses how to model a merry-go-round (well, really with just one element hanging the mechanism is often called a “rotational pendulum”) using the Euler-Lagrange formalism.

The first ten minutes review the kinematics (defining the coordinates that describe the motion plus deriving expressions of positions, speeds and accelerations of the elements of the system based on said coordinates). The `jacobian` command is widely used to apply the “chain rule”. If the manipulations are unfamiliar to you, perhaps you should preview the video [??].

Next, the Euler Lagrange equations are expressed with the Symbolic toolbox and the meaning of the different terms is analyzed, grouping them in the usual way $M(q)\ddot{q} = \tau + C(q, \dot{q})\dot{q} + G(q)$, i.e., with mass matrix and Coriolis terms.

Matlab code for simulations will be discussed in next video [55], and further digression on particular cases of interest will be discussed in video [54].

[54: tiovEL2EN]

2DoF dynamics of a carousel via Euler-Lagrange equations:
particular cases
Materiales: [[CÓD.:](#) tiovivo2GLEulerEN.mlx] [[PDF](#)]

***  13:15
[\[YouTube ▶\]](#)

This video presents particular cases of the equations of motion of a merry-go-round carousel (with a single passenger) with two degrees of freedom, which were developed in the video [??].

In particular, the video discusses three situations:

- Zero Suspended Length
- Spherical pendulum (without top platform, load hanging from the center)
- Analysis of lateral dynamics only (this is not actually a particular case, but simply studying one of the two equations of motion in isolation).

Motivated by the latter case, the final part of the video discusses the idea of the *computed torque* necessary for the top platform to follow a certain prescribed trajectory, used in dynamic control of robotic mechanisms.

[55: tiovELsimEN] 2DoF dynamics of a carousel: simulation and animation
12:43

*** ☰

Materiales: [[Cód.:](#) tiov2ganimv2.m] [[PDF](#)]

[YouTube ▶]

This video animates the equations of motion of a merry-go-round (with a single hanging element) that were derived in the video [53] using the Euler-Lagrange methodology.

To do this, the results of the aforementioned video that used the Symbolic Toolbox are transformed/compiled into “numeric” code (i.e., non-symbolic ordinary Matlab functions), using `matlabFunction`, and everything is prepared in the required format to execute `ode45`. Different `plot` and `plot3` are used to draw each animation frame.

2.2.5 Dinámica longitudinal de vuelo de aeronave (fugoide)

[56: intrid1] Dinámica del movimiento de un punto (movimiento plano 2GL)
en coordenadas intrínsecas T, N (tangencial, normal)

** 👤 15:29

Materiales: [FrenetCdGDinamica.pdf]

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute la obtención de las ecuaciones en coordenadas intrínsecas (tangente, normal) del movimiento de un punto sujeto a fuerzas. Si la dirección del vector velocidad es $\vec{T} := (\cos \theta, \sin \theta)$, entonces $\vec{v} = \nu \vec{T}$, siendo ν la velocidad del “velocímetro” del coche (esto es, ν es un escalar). Entonces $\frac{d\nu}{dt} = \frac{F_T}{m}$, y $\frac{d\theta}{dt} = \frac{F_{NL}}{m\nu}$ siendo F_T la fuerza tangencial, y F_{NL} la fuerza normal (positiva si estira hacia la izquierda de la trayectoria). La demostración de estas expresiones y la discusión de su significado son el objetivo del vídeo.

La parte final recuerda que un sólido rígido tiene un grado de libertad extra (angular) y que por tanto no tiene que estar orientado con el ángulo θ de la trayectoria del centro de masas, poniéndose como ejemplo maniobras de avión o de coche deportivo. Las ecuaciones con ese grado de libertad adicional no se contemplan, por brevedad, en este vídeo introductorio, que sólo aborda la dinámica de una “masa puntual”: consideramos que “path frame” y “body frame” son idénticas.

[57: fugoid1] Modelado de planeo fugoide longitudinal de una aeronave (EDO
2o orden simplificada)

*** ☰ 15:16

Materiales: [PhugoidIntroES.pdf]

[YouTube ▶]


* [LINK TO ENGLISH VERSION](#)

Este vídeo presenta el modelado del vuelo “fugoide” de una aeronave. Bajo ciertas suposiciones (movimiento plano, inercia rotacional despreciable, gran radio de curvatura de trayectoria), se puede considerar al avión como una masa puntual sujeta a fuerzas tangenciales y normales, y escribir las ecuaciones del movimiento en coordenadas intrínsecas, discutidas en el vídeo [56]. Aquí se trata simplemente de sustituir en dichas ecuaciones de movimiento el empuje del motor y la resistencia aerodinámica en la dirección tangencial, la sustentación en la dirección normal, y el peso descomponerlo en sus componentes tangencial y normal. Con ello se pueden escribir las ecuaciones de Zhukovski/Lanchester que aproximan este modo de la dinámica de vuelo que se denomina “fugoide”.

Nota: el empuje se ha supuesto una entrada constante, pero en algunos casos (p.ej. motores de hélice) puede depender de la velocidad aerodinámica (airspeed). No se ha considerado esto (y muchas otras cosas) en estos materiales, por simplicidad.

[58: fugsim]

Planeo fugoide de aeronave (ecs. simplificadas): ejemplos de simulación/animación y discusión

***  10:39

Materiales:

[[CÓD.:](#) animaavion.m] [[PDF](#)]

[YouTube ▶]


*[LINK TO ENGLISH VERSION](#)

Este vídeo presenta simulaciones con distintos coeficientes de sustentación / arrastre y valores de empuje del modelo de la dinámica fugoide simplificado cuya obtención se discutió en el vídeo [57]. El detalle del código Matlab que ejecuta las simulaciones y animaciones se presenta en el vídeo [59].

Aquí se discuten: planeo descendente, vuelo nivelado horizontal, ascenso y ascenso inestable, que ocurre si el empuje sube por encima de cierto valor. Detalles sobre puntos de equilibrio y estabilidad se discuten en el vídeo [118].

[59: fugsimcod]

Planeo fugoide de aeronave (ecs. simplificadas): código de simulación ode45 y animación (Matlab)

***  09:43

Materiales:

[[CÓD.:](#) animaavion.m] [[PDF](#)]

[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo presenta el detalle del código Matlab necesario para hacer simulaciones y animaciones del vuelo fugoide 2D (dinámica longitudinal) de una aeronave. Los primeros dos minutos revisan las ecuaciones de estado a simular (detalles en vídeo [57]). Un análisis teórico de los puntos de equilibrio y su linealización aparecen en el vídeo [118].

Luego, se discute cómo introducir el modelo, cómo llamar correctamente a `ode45` y, en la parte final, se discute cómo representar la respuesta como


1. una gráfica de posiciones, velocidades, aceleraciones, etc. en función del tiempo;
2. una animación del movimiento del avión

La ejecución de diferentes simulaciones con diferentes parámetros aerodinámicos, velocidades iniciales o empuje se aborda en detalle en el vídeo [58].

2.2.6 Caso de estudio: móvil restringido en un alambre

[60: mcm1]

Ejemplo modelado de sistema mecánico bola guiada (1): cinemática

**  12:26

Materiales:

[[CÓD.:](#) MountainCarModelling.mlx] [[PDF](#)]

[YouTube ▶]

Este vídeo plantea el problema de modelar el movimiento de una bola agujereada que pasa a través de un alambre cuya forma viene dada como $y = f(x)$.

Este primer vídeo discute el planteamiento del problema, la notación y la cinemática (ecuaciones de posición, velocidad, aceleración, aceleración tangencial y aceleración normal). Podría ser de interés visionar previamente el vídeo [915] que revisa derivadas parciales, jacobianos, derivadas totales, etc. o el ejemplo matlab en el vídeo [916].

En el caso de haber “looping” (vueltas hacia atrás, montaña rusa), no se podría expresar la posición como $y = f(x)$. La cinemática de ese caso se analiza en los dos primeros minutos del vídeo [66].

Las ecuaciones del movimiento dinámicas se discuten en los vídeos [61] (Newton, balance de fuerzas), [62] (Euler-Lagrange 1ª forma, implícita), [63] (Euler-Lagrange 2ª forma, paramétrica). La simulación de casos concretos `ode45` se aborda en el vídeo [64].

[61: mcm2] **Ejemplo modelado de sistema mecánico bola guiada (2):**
dinámica (Newton, balance de fuerzas)

Materiales: [[CÓD.:](#) MountainCarModelling.mlx] [[PDF](#)]

** 11:41

[YouTube ▶]

Este vídeo plantea el problema de modelar el movimiento de una bola agujereada que pasa a través de un alambre cuya forma viene dada como $y = f(x)$.

Los dos primeros minutos discuten el planteamiento del problema, la notación y la cinemática, discutidos en detalle en el vídeo [60].

Las ecuaciones del movimiento dinámicas se desarrollan mediante dos métodos equivalentes:

1. Balance de fuerzas tangenciales a la trayectoria
2. Balance de fuerzas horizontal y vertical

El primero de ellos es más sencillo (no aparece la reacción, el modelo tiene solamente una ecuación), pero en el segundo de ellos permite obtener una expresión explícita de la fuerza de reacción que podría ser de utilidad en algunas aplicaciones (resistencia de materiales, adherencia a la trayectoria en caso de solamente apoyo de modo que la reacción sólo pueda tener un sentido, etc.).

Otras metodologías de obtención de las ecuaciones del movimiento se discuten en los vídeos [62] (Euler-Lagrange 1ª forma, implícita), [63] (Euler-Lagrange 2ª forma, paramétrica). La simulación de casos concretos [ode45](#) se aborda en el vídeo [64].

[62: mcm3el1] **Ejemplo modelado de sistema mecánico bola guiada (3):**
dinámica (Euler-Lagrange, forma 1)

Materiales: [[CÓD.:](#) MountainCarModelling.mlx] [[PDF](#)]

**** 11:15

[YouTube ▶]

Este vídeo plantea el problema de modelar el movimiento de una bola agujereada que pasa a través de un alambre cuya forma viene dada como $y = f(x)$.

Los tres primeros minutos discuten el planteamiento del problema, la notación y la cinemática y balances de fuerzas, discutidos en detalle en los vídeos [60] y [61].

Las ecuaciones del movimiento dinámicas se desarrollan mediante las ecuaciones de Euler-Lagrange en la forma en la que incorporan un multiplicador de Lagrange λ que multiplica a la restricción $y - f(x) = 0$ y sus derivadas parciales.

Otras metodologías de obtención de las ecuaciones del movimiento se discuten en los vídeos [63] (Euler-Lagrange 2ª forma, paramétrica). La simulación de casos concretos [ode45](#) se aborda en el vídeo [64].

[63: mcm4el2] **Ejemplo modelado de sistema mecánico bola guiada (4):**
dinámica (Euler-Lagrange, forma 2, paramétrica)

Materiales: [[CÓD.:](#) MountainCarModelling.mlx] [[PDF](#)]

**** 13:58

[YouTube ▶]

Este vídeo plantea el problema de modelar el movimiento de una bola agujereada que pasa a través de un alambre cuya forma viene dada como $y = f(x)$.

El primer minuto discuten el planteamiento del problema, la notación y la cinemática, discutidos en detalle en el vídeo [60].

Las ecuaciones del movimiento dinámicas se obtienen mediante la metodología Euler-Lagrange escogiendo como coordenada generalizada el desplazamiento horizontal x .


Otras metodologías de obtención de las ecuaciones del movimiento se discuten en los vídeos [61] (Newton, balances de fuerzas) y [62] (Euler-Lagrange 1ª forma, implícita);

todas obtienen el mismo resultado. La simulación de casos concretos `ode45` se aborda en el vídeo [64].

También es aconsejable ver el resumen rápido de los tres primeros minutos del vídeo [65] donde se muestra el código Euler-Lagrange mínimo (sin toda la parafernalia necesaria para deducir y comparar las ecuaciones de movimiento por otros métodos), comprobando que en efecto se puede modelar la dinámica del sistema con una docena de líneas de código.

Un caso muy parecido pero con “looping” de modo que ya no se puede expresar y como una función de x , esto es, una montaña rusa, se aborda en el vídeo [66].


[64: mcm5sim] Ejemplo modelado de sistema mecánico bola guiada (5):
simulación ode45
Materiales: [Cód.: `simulaAnimaMountaincar.m`] [PDF]

***  10:27
[YouTube ▶]

En este vídeo se simula con `ode45` la ecuación del movimiento de una masa puntual (bola ensartada en un alambre) sujeta a $y = f(x)$. Realmente son una docena de líneas de vídeo una vez se tienen las ecuaciones de la aceleración \ddot{x} , obtenidas por varios métodos en los vídeos [61] (Newton, balance de fuerzas), [62] (Euler-Lagrange 1ª forma, implícita), [63] (Euler-Lagrange 2ª forma, paramétrica).

Un caso muy parecido con “looping” de modo que ya no se puede expresar y como una función de x , esto es, una montaña rusa, se aborda en el vídeo [66].

[65: mcm6ha] Ejemplo modelado de sistema mecánico bola guiada (6):
Hamiltoniano
Materiales: [Cód.: `MountainCarModellingONLYHamilton.mlx`] [PDF]

*****  12:37
[YouTube ▶]


Este vídeo plantea el modelado dinámico de una bola ensartada en un alambre $y = f(x)$ con las ecuaciones de Hamilton.

Se forma el Hamiltoniano a partir del Lagrangiano y se plantean las dos ecuaciones de la derivada del momentum generalizado y de la coordenada que describe el movimiento (en este caso x).

Se recomienda, para una mejor comprensión del presente material, el visionar previamente los vídeos [60], [61], [62], [63] y la simulación `ode45` en [64] si no se está familiarizado con el caso de estudio.

Montaña Rusa

[66: rollerco] Modelado y simulación montaña rusa (bola guiada con looping)
por Newton y por Euler-Lagrange
Materiales: [Cód.: `RollerCoasterModelaSimula.zip`] [PDF]

****  11:24
[YouTube ▶]

Este vídeo modela (por balance de fuerzas y por Euler-Lagrange) las ecuaciones de una “montaña rusa” o “roller-coaster” unidimensional, donde un móvil describe una curva dada por $r := (x(q), y(q))$ siendo q un parámetro geométrico para poder dibujarla. La curva tiene “loopings” (da la vuelta) y por tanto no puede ser expresada como $y = f(x)$.

El desarrollo es muy parecido al de los vídeos del caso de estudio de bola en alambre $y = f(x)$, por lo que aquí se ha hecho de forma muy escueta. Para una mejor comprensión, se recomienda visualizar previamente dicho caso de estudio, en los vídeos [60], [61], [63], [64].

Otro ejemplo sencillo de modelado Euler-Lagrange con restricciones de un sistema de 1GL aparece en el vídeo [??].

2.2.7 Sistema Bola-carrito de 2 grados de libertad

[67: cpNewton]

Ejemplo modelado de sistema carrito-bola (1): cinemática y balances de fuerzas
Materiales: [[CÓD.:](#) CartPole.mlx] [[PDF](#)]

***  14:59
[\[YouTube ▶\]](#)


Este vídeo describe la cinemática y modela las ecuaciones dinámicas de cuarto orden de un sistema bola-carrito (2 grados de libertad), sin rozamiento y suponiendo bola de masa puntual en el extremo (el caso con rozamiento e inercia será objetivo de otros materiales).

Las ecuaciones de la dinámica se obtendrán haciendo balances de fuerzas resultantes (Newton). Con las ecuaciones de Euler-Lagrange descritas en el vídeo [68] se llega a un modelo totalmente idéntico.

Un caso más general donde el momento de inercia de la masa que gira no sea despreciable se aborda en el vídeo [69], y la simulación (con momento de inercia y rozamiento) se aborda en el vídeo [70] aunque adaptarlo al código aquí desarrollado sólo requiere de cambios triviales.

[68: cpEL]

Ejemplo modelado de sistema carrito-bola (2): dinámica mediante Euler-Lagrange
Materiales: [[CÓD.:](#) CartPole.mlx] [[PDF](#)]


***  10:13
[\[YouTube ▶\]](#)

En este vídeo se obtienen las ecuaciones dinámicas de cuarto orden de un sistema bola-carrito (2 grados de libertad), sin rozamiento y suponiendo bola de masa puntual en el extremo con las ecuaciones de Euler-Lagrange. El resultado, como era de esperar es el mismo modelo que el obtenido con balances de fuerzas en el vídeo [67].

Un caso más general donde el momento de inercia de la masa que gira no sea despreciable se aborda en el vídeo [69]. La simulación numérica y animación se discuten en el vídeo [70]. Un ejemplo Euler-Lagrange de péndulo con múltiples eslabones se aborda en los vídeos [74] y [75].

[69: cpine]

Ejemplo modelado de sistema carrito-bola con momento de inercia en péndulo no despreciable
Materiales: [[CÓD.:](#) CartPoleWithRotInertia.mlx] [[PDF](#)]


****  15:17
[\[YouTube ▶\]](#)

En este vídeo se obtienen las ecuaciones del movimiento de un sistema carrito-barras donde sí se considera el posible momento de inercia de la masa que oscila colgada. De este modo, el caso con inercia despreciable considerado en los vídeos [67] y [68] será un caso particular del ejemplo aquí detallado. Se considera la obtención de las ecuaciones del movimiento tanto por Euler-Lagrange como con balances de fuerzas y pares (Newton).

El vídeo [70] utiliza `ode45` para simular (y presentar una animación) numéricamente el movimiento del sistema dados unos ciertos valores de los parámetros constantes.

[70: cpinesim]

Simulación ode45 sistema carro-péndulo 2GL
Materiales: [[CÓD.:](#) simulaAnimaCartPole.m] [[PDF](#)]

***  15:28
[\[YouTube ▶\]](#)

En este vídeo se simulan las ecuaciones del movimiento del sistema carro-bola (bueno, carro + péndulo sólido rígido con momento de inercia no despreciable) con `ode45`. El detalle de cómo obtener las ecuaciones del movimiento se aborda en el vídeo [69], aunque realmente no es necesario para entender los conceptos de simulación aquí presentados; una

versión más sencilla (sin momento de inercia ni rozamiento) del modelado se plantea en el vídeo [67].

A partir de unas expresiones de las aceleraciones a_x , a_θ , se plantea el modelo de orden 4: $\dot{x} = v_x$, $\dot{\theta} = v_\theta$, $\dot{v}_x = a_x$, $\dot{v}_\theta = a_\theta$; se sustituyen parámetros constantes, se compila de simbólico a numérico con `matlabFunction` y con ello se puede ejecutar `ode45`. Se simulan varios escenarios con/sin rozamiento, con/sin fuerza externa, o con varias tolerancias numéricas para el simulador. La parte final del vídeo describe brevemente el código para hacer una animación.

Otro código de animación de un péndulo/robot de múltiples eslabones se aborda en el vídeo [76]; el modelado de dicho robot se desarrolla en los vídeos [74] y [75].

[71: cpforced] Ejemplo modelado sistema carro-bola con movimiento del carro forzado **** 16:58
Materiales: [[CÓD.:](#) CartPoleForced.mlx] [[PDF](#)] [YouTube ▶]

Este vídeo plantea el modelado dinámico de un sistema carro-bola idéntico al considerado en los vídeos [67] y [68], pero asumiendo que el desplazamiento del carrito está “forzado” debido a dos posibles causas tecnológicas (que son irrelevantes desde el punto de vista del código que resuelve el problema):

- que el carrito forme parte de otro mecanismo de mucha masa de modo que el efecto del péndulo (fuerza de reacción) sobre su movimiento sea despreciable.
- que la trayectoria forzada sea una trayectoria “deseada” de modo que queramos calcular las fuerzas sobre el carrito que darían lugar a dicha trayectoria (problema de control).

Por completitud, los primeros cuatro minutos del vídeo plantean el problema y revisan brevemente el modelado Euler-Lagrange, pero el público poco familiar con dicha metodología debería visualizar el vídeo [68] previamente a éste.

Se plantean dos opciones para resolver el problema:

- resolver el modelo completo 2GL pero en vez de despejar aceleraciones a_x y a_θ como en los citados vídeos, se despeja a_θ y la fuerza excitadora F necesaria.
- Plantear un lagrangiano dependiente del tiempo (reónómico) y usar ecuaciones Euler-Lagrange sólo sobre el grado de libertad angular dado que el otro está forzado. Los desarrollos deberán tener en cuenta tanto la dependencia implícita del tiempo como la explícita (vídeos [915] y [916]) en la cinemática y el lagrangiano.

Ambos métodos obtienen el mismo resultado.

2.2.8 Sistema catapulta barra-bola

[72: ballbar2GL] Modelado dinámico Euler-Lagrange de una catapulta (bola en barra) *** 13:08
Materiales: [[CÓD.:](#) BolaEnBarra.mlx] [[PDF](#)] [YouTube ▶]

En este vídeo se modela la cinemática y dinámica de una barra de una cierta inercia con una masa puntual (bola agujereada ensartada en la barra) que es “catapultada” hacia el exterior si no está en el centro (eje de rotación).

Se utilizan las ecuaciones de Euler-Lagrange de 2 grados de libertad, y se plantean las ecuaciones del movimiento incluyendo término de gravedad (giro en un plano vertical), aunque obviamente haciendo $g = 0$ se tienen las ecuaciones de una catapulta horizontal.

El vídeo [73] analiza el caso particular de catapulta donde la barra es “forzada” a girar con una velocidad constante.

[73: ballbar1GL]

Modelado dinámica Euler-Lagrange de catapulta (bola en barra) con rotación forzada 1GL


11:48

Materiales:
[
CÓD.: BolaEnBarra.mlx
]
[
PDF
]

[YouTube ▶]

En este vídeo se modela la cinemática y dinámica de una barra con una masa puntual (bola agujereada ensartada en la barra) que es “catapultada” hacia el exterior si no está en el centro (eje de rotación).


En este caso, la velocidad angular de la barra se supondrá constante (forzado) y se obtendrán ecuaciones considerando un único grado de libertad (aunque, claro, las ecuaciones de este movimiento forzado se podrían obtener como caso particular de las ecuaciones de dos grados de libertad consideradas en el vídeo [72], cuya visualización se recomienda).

Se utilizan tanto los balances de fuerzas como las ecuaciones de Euler-Lagrange de 1 grado de libertad (lagrangiano dependiente del tiempo).

2.2.9 Péndulo de múltiples eslabones

[74: pendNcin]

Modelo dinámico de Péndulo/Robot de múltiples eslabones (I):
cinemática y energías


16:37


Materiales:
[
CÓD.: ModelPend.mlx
]
[
PDF
]

[YouTube ▶]

Este vídeo plantea el problema de un robot articulado (péndulo) de múltiples eslabones de N grados de libertad, introduciendo notación, discutiendo su geometría, la cinemática de velocidades y el cálculo de energías cinética y potencial. Las ecuaciones Euler-Lagrange del movimiento serán obtenidas en el vídeo [75], y la simulación y animación en el vídeo [76].

[75: pendNEL]

Modelo dinámico de Péndulo/Robot de múltiples eslabones (II):
ecuaciones Euler-Lagrange


13:58

Materiales:
[
CÓD.: ModelPend.mlx
]
[
PDF
]


[YouTube ▶]

Este vídeo plantea las ecuaciones Euler-Lagrange de la dinámica de un robot (péndulo) de múltiples eslabones de N grados de libertad; aunque se revisa aquí en los primeros dos minutos de vídeo, el vídeo [74] introduce notación, geometría, cinemática y el cálculo de energías cinética y potencial. La simulación y animación con las ecuaciones de movimiento obtenidas se discuten en el vídeo [76].

Nota: casos más sencillo de sistemas de 2 grados de libertad se abordan en los vídeos [68] y [69], por ejemplo. Quizás fuera conveniente visualizarlos previamente a este contenido.

[76: pendNSim]

Modelo dinámico de Péndulo/Robot de múltiples eslabones
(III): simulación y animación


20:05

Materiales:
[
CÓD.: simulPend.m
]
[
PDF
]


[YouTube ▶]

En este material se presenta el código de simulación de las ecuaciones dinámicas obtenidas en el vídeo [75] para un mecanismo de varios eslabones articulados. Primero, se comprueba que con un único grado de libertad las ecuaciones son las de un péndulo simple. Luego, se sustituyen datos numéricos y se hacen diversas simulaciones cambiando el número de eslabones, el rozamiento y la masa colgada en el extremo libre. Un conjunto de instrucciones “plot” es usado también para hacer una animación, similar a la de, por ejemplo, el vídeo [70]. Si estás interesado en código de animación, un ejemplo de varias masas y muelles aparece en el vídeo [35].

2.3 Modelado basado en estadística: ejemplo biorreactor

[77: bio1mod] Sistema inestable 1er orden: modelo bioreactor microscópico (estadística) y macroscópico (ec. diferencial)

Materiales: [[CÓD.: PIDcontrolAPP1.0.zip](#)]

******  **19:53**

[YouTube ▶]

Este vídeo describe el modelado de un bioreactor como un sistema de primer orden inestable, de modo que el modelo predice un crecimiento exponencial de la población de un microorganismo, a partir de las conclusiones de un modelo ‘microscópico’ estadístico sobre la ‘probabilidad de división de una célula en un intervalo de tiempo pequeño’. El resultado es un modelo ‘macroscópico’ de tipo $\frac{dx}{dt} = 0.4x$ (ecuación diferencial ordinaria), con solución $x(t) = e^{0.4t}x(0)$. La base estadística es análoga a la del ‘decaimiento radiactivo exponencial’ en física, como se sugiere en el vídeo.

Si además del crecimiento ‘natural’ por división celular existe un aporte de biomasa u (o extracción si el signo de u fuese negativo), entonces el modelo será $\frac{dx}{dt} = 0.4x + u$. Este modelo sigue siendo inestable en lazo abierto: para mantener una población constante de microorganismos deberá ser ‘estabilizado’ mediante control en bucle cerrado (no objetivo de este vídeo, que únicamente aborda modelado).

Hay que remarcar que este modelo es muy aproximado: el crecimiento de población consume nutrientes, aparecen metabolitos secundarios que a veces son tóxicos, hay mutaciones, etc. de modo que solamente es una primera aproximación a las consideraciones en este tipo de sistemas y también sirve de motivación al análisis de sistemas inestables en control industrial, dado que los sistemas usuales de circuitos eléctricos, fluidos, electromecánicos, etc. en asignaturas iniciales de automática suelen ser estables.

Por ejemplo, como inciso, los crecimientos exponenciales con tasa de crecimiento que es ella misma aleatoria, bajo ciertas asunciones adicionales, dan lugar a los modelos Black-Scholes-Merton en economía.

2.4 Modelos de parámetros distribuidos (medio continuo)

Existen modelos donde su dinámica se rige por ecuaciones diferenciales en derivadas parciales (EDP). Este tipo de modelos se origina conceptualmente al suponer que las vibraciones, la transmisión de calor, el campo magnético, etc. en un objeto están caracterizados por una deformación, velocidad, temperatura, campo eléctrico, etc. en cada *punto* de la geometría de dicho objeto. Por ejemplo, la difusión del calor en un objeto tendría una forma $T(x, t)$ siendo $x \in \mathbb{R}^3$ un punto perteneciente al objeto (coordenada espacial) y siendo t la variable que representa al tiempo.

Las ecuaciones en derivadas parciales aparecen conceptualmente al aplicar leyes físicas tras dividir el cuerpo en “fragmentos” o “elementos” más y más pequeños... hasta que se hacen “infinitesimales”. Sólo en muy pocos problemas se puede obtener una solución explícita de dichas ecuaciones en derivadas parciales y, en ingeniería, se busca obtener modelos de una complejidad (orden) razonable para cada aplicación. Estos modelos pueden ser obtenidos replicando esa metodología conceptual de dividir el objeto en “fragmentos” (discretización espacial), pero parando cuando el número de elementos se considere suficiente para describir las propiedades que se buscan (y para poder manejarlo en el adecuado código de modelado y simulación). Otra metodología, denominada “discretización por diferencias finitas” sustituye derivadas parciales por aproximaciones, por ejemplo la de Euler $\frac{\partial T(x, t)}{\partial t} \approx \frac{T(x, t+h) - T(x, t)}{h}$, o $\frac{\partial T(x, t)}{\partial x} \approx \frac{T(x+h, t) - T(x, t)}{h}$, y también hay métodos de “elementos finitos” donde se transforman ecuaciones a una forma integral (*weak form*), se asume cierta función de interpolación, etc.

Estos materiales no tienen como objetivo el ilustrar en profundidad y con rigor todos los fenómenos físicos descritos por EDP. Se remite al lector a los adecuados libros de texto de Matemáticas (si se quiere explorar formalmente sus propiedades y soluciones), de Cálculo Numérico (si se desea comprender las metodologías numéricas de solución por elementos finitos, diferencias finitas, colocación

ortogonal, etc.), y de Ingeniería Mecánica, Transmisión de Calor, Electromagnetismo, Mecánica de Fluidos, etc. para el detalle de los problemas tecnológicos que dicho tipo de modelos contribuyen a solucionar.

El objetivo de esta Sección es presentar ejemplos sencillos, en un lenguaje entendible en el contexto de estudiantes del ámbito del Control Automático, para obtener modelos $\dot{x} = Ax + Bu$ de orden finito que aproximen sistemas modelados por EDP. La solución no tiene por que ser la mejor técnicamente considerada entre los expertos en la especialidad mecánica, eléctrica, etc.

[78: femod] Modelos de parámetros distribuidos: modelado de la vibración longitudinal de un muelle (discretización espacial)

Materiales: [[CÓD.:](#) ModeladoMuelleOndaLongitudinal.mlx] [[PDF](#)]

*** 10:47

[[YouTube](#) ►]

Este vídeo discute el modelado de las vibraciones longitudinales de un muelle con masa no despreciable mediante su división en N sistemas físicos idénticos “masa-muelle-amortiguador” más pequeños (discretización espacial). Cuando los elementos se hacen infinitesimales, el resultado es la ecuación de onda unidimensional, modelando de la propagación con una cierta velocidad de ondas longitudinales como se ve en https://www.youtube.com/watch?v=_DOCSUa0sbM. Este vídeo sería el llevar “hasta el infinito” los modelos de varias masas y muelles como el discutido en los vídeos [24] y [25], cuya visualización previa a éste se recomienda si no estas muy familiarizado con los conceptos básicos de modelado.

En el caso de este vídeo, se planteará cómo modelar cuando se ha escogido un valor de N para obtener una representación interna normalizada $\dot{x} = Ax + Bu$ de orden $2N$. Nótese que la solución podría ser diferente a la que un experto en Ingeniería Mecánica (elasticidad y resistencia de materiales) podría proponer, usualmente basada en el módulo de Young del material.

Nota: la simulación en tiempo y frecuencia del modelo aquí construido se aborda en el vídeo [278], una vez se hayan introducido los conceptos adecuados.

[79: termedp] Modelado unidimensional de un calentador tubular de líquido con ecuaciones en derivadas parciales (EDP)

Materiales: [HeatExchModelEDP.pdf]

**** 10:56

[[YouTube](#) ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo modela un calentador tubular con una resistencia calefactora como una conexión de múltiples elementos donde cada uno de los elementos es un mini-tanque calefactor de primer orden idéntico al modelado en el vídeo [27]; este modelado se revisa rápidamente en el primer minuto y medio del vídeo, para que sea más autocontenido.

Tras plantearse las relaciones entre las salidas de un elemento y las entradas del siguiente, se pasa a considerar que los elementos estarán separados una distancia dx , y que el volumen de cada elemento será Sdx . Conceptualmente, si la temperatura de un elemento es $T(x)$, la temperatura del siguiente elemento será $T(x + dx)$. Como estamos analizando cambios de temperatura en el tiempo (dinámica), realmente se trata de ver las ecuaciones que rigen los valores $T(x, t)$ en distintas posiciones e instantes de tiempo.

Planteando las ecuaciones de cada elemento y haciendo el límite cuando $dx \rightarrow 0$, se obtiene la ecuación en derivadas parciales que describe al calentador tubular. Tiene la forma:

$$\frac{\partial T}{\partial t} = -\frac{1}{S}F\frac{\partial T}{\partial x} - \frac{\bar{\kappa}}{S\rho C_e}T + \frac{\bar{Q}}{S\rho C_e}$$

La parte final del vídeo discute dos casos particulares muy estudiados.

El primero es el caso de no disipación/generación $\bar{\kappa} = 0$, $\bar{Q} = 0$. Con ello, la ecuación se convierte en la EDP de transporte $\frac{\partial T}{\partial t} = -v\frac{\partial T}{\partial x}$ siendo $v = F/S$ la velocidad de circulación

del fluido; su solución es un retardo (denominado retardo de transporte) que depende del valor del caudal F .

El segundo caso particular es el caso estacionario (equilibrio, $\frac{\partial T}{\partial t} = 0$) que consideraremos por simplicidad sin aporte de calor por la resistencia, $\dot{Q} = 0$. En ese caso, al eliminar las derivadas temporales, se obtiene una ecuación diferencial ordinaria en la longitud $\frac{\partial T_{eq}}{\partial x} = -\frac{\bar{\kappa}}{F\rho C_e} \cdot T_{eq}$ cuya solución exponencial da lugar a fórmulas muy utilizadas en cálculos de intercambiadores de calor, no objetivo de este material (la fórmula exponencial es utilizada en el vídeo [29] para obtener una aproximación sencilla, de primer orden, de la dinámica del intercambiador que al menos preserve el balance de energía correcto en equilibrio).

Si no se puede calcular analíticamente la solución exacta en el caso general, la simulación numérica requiere, en la mayor parte de casos, de modelos de orden finito (elementos no infinitesimales) como se había hecho en las fases iniciales del modelado aquí presentado: en efecto, las EDP tienen un número infinito de estados que no permiten su simulación con modelos $\dot{x} = f(x, u)$ manejables en un computador.

En este caso, para caudal F constante *la solución exacta sí puede calcularse*, ver vídeo [184], donde se hace por transformada de Laplace. La simulación comparada de las aproximaciones de primer orden detalladas en otros vídeos y la solución EDP exacta se aborda en el vídeo [186].

Capítulo 3

Linealización

Como las ecuaciones algebraicas (y algebraico-diferenciales) no lineales son “difíciles” de resolver (tanto analíticamente como numéricamente), en la base de muchas metodologías en matemáticas e ingeniería está la aproximación “local” a una función lineal, que resulta mucho más fácil de manejar tanto numéricamente como teóricamente. Este capítulo discute esas ideas.

3.1 Funciones de una variable

[80: lint1] Linealización de funciones de una variable (recta tangente)

Materiales: [lnteoria1.pdf]

*  10:57
[YouTube ►]

Este video presenta el concepto de aproximación lineal a una curva mediante la expresión de su recta tangente, en funciones reales de una variable.

Define el concepto de función lineal (recta que pasa por el origen, proporcionalidad), cómo calcular la recta tangente a una curva en un punto (mediante la derivada), y el cambio de variable incremental para transformar el punto de tangencia en el origen.

Presenta un ejemplo con $y = \sqrt{16 + x^2}$, comparando la salida de esta función con su linealización alrededor de $x_0 = 3$.

Interpretaciones adicionales y observaciones de interés se abordan en el vídeo [81], continuación de éste, así como en el ejemplo basado en el Symbolic Toolbox de Matlab desarrollado en el vídeo [82].

[81: lint2] Linealización de funciones de 1 variable (II): serie de Taylor, discusión, conclusiones

Materiales: [lnteoria2taylor1vb.pdf]

*  12:18
[YouTube ►]

En este vídeo, se presenta una alternativa al concepto geométrico de “recta tangente” para linealizar, utilizando los términos de primer orden de la Serie de Taylor (generalizable a polinomio tangente de grado arbitrario).

Se presenta ejemplo de linealización de $y = \sin(x)$ en distintos puntos. Un ejemplo similar ($f(x) = \cos(x)$) usando la Symbolic Toolbox de Matlab se aborda en el vídeo [82].


Se presenta, como ejemplo de aplicación de la linealización, la resolución de ecuaciones (estáticas) no lineales: el resultado es el *método de Newton*.

Existen otros contextos donde se aproximan funciones por rectas (acotación, regresión, ...). Se discute las diferencias con la linealización.

El caso multivariable se presenta en el vídeo [83], y los sistemas dinámicos en el [86].

[82: linsisoml]

Linealización de funciones de 1 variable: ejemplo Matlab
 (Symbolic Toolbox)
Materiales: [[CÓD.:](#) LinealizaSISOejemplo.mlx] [[PDF](#)]

**  12:40
 [YouTube ▶]

Este vídeo presenta un ejemplo Matlab (aprovechando también para revisar los conceptos básicos) de las ideas de los vídeos [80] y [81].


En concreto, linealiza $f(x) = \cos(x)$ en $x_0 = 0.85$, dibujando la función y la recta tangente (desarrollo de Taylor hasta primeras derivadas). La parte final del vídeo analiza cómo representar gráficamente el intervalo alrededor de x_0 donde el error del modelo linealizado está por debajo del 10%.

La versión multivariable de este ejemplo se aborda en el vídeo [84].

3.2 Aproximación de funciones de varias variables

[83: lint3]

Linealización (III): caso multivariable (subespacio tangente)
Materiales: [lnteoriaMV.pdf]

** / ***  10:49
 [YouTube ▶]

En este video se generaliza la idea de la recta tangente al caso de varias variables en una expresión $f(x_1, \dots, x_n) = 0$.

*Aquí se aborda un enfoque *geométrico* de “tangente” pero, como en el caso de funciones de una variable, también hay un enfoque basado en la serie de Taylor multivariable que se discute en el vídeo [84], cuya visualización también se aconseja para entender las dos formas de abordar el problema.

Se definen las funciones lineales de varios argumentos. El objetivo es aproximar una función no lineal de varias variables a ese tipo de funciones.

*** Se define el plano tangente (2 argumentos) y la “variedad tangente” para el caso general. Una ampliación de dichos conceptos, fuera de los objetivos de este material introductorio, puede consultarse en los vídeos [917] y [918] ****.

Se introduce la matriz jacobiana de primeras derivadas parciales (prerequisitos sobre jacobianos, derivadas parciales y totales en vídeos [915] y [916] ***).

Se discute el ejemplo numérico de linealizar la curva implícita $2 - (0.2x_1^2 + 0.1x_1x_2 + x_2^2) = 0$.

También se discute el linealizar un sistema (estático) descrito por dos ecuaciones en tres variables.

Otro ejemplo detallado (incluyendo revisión teórica), utilizando la symbolic toolbox de Matlab, se aborda en el vídeo [84].

[84: linmiso1]

Linealización de función de 2 variables; ejemplo Matlab
 (symbolic toolbox + revisión teórica Taylor)
Materiales: [[CÓD.:](#) LinealizaMISOejemplo.mlx] [[PDF](#)]

**  16:54
 [YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo desarrolla un ejemplo Matlab de la linealización de una función de 2 variables $f(x_1, x_2) = x_1^2/2 + x_2^2 + \cos(x_1 + e^{-x_2})$ en un punto de funcionamiento arbitrariamente elegido.

Puedes consultar un material dedicado exclusivamente a la teoría en el vídeo [83] que quizás debas visualizar junto a este vídeo, donde se presenta un enfoque “geométrico” (variedad/subespacio tangente) complementario al aquí discutido (serie de Taylor).

Se define y representa la función, se hace una revisión teórica del desarrollo de Taylor multivariable y de las definiciones de Jacobiano y Hessiano, y se aplica al ejemplo concreto para linealizar usando el comando `jacobian`.

La parte final del vídeo superpone los gráficos (3D) de la función original y su aproximación linealizada, comprobando que se trata del plano tangente.

El vídeo [85], opcional, evaluará la exactitud de la aproximación en un punto numérico concreto y también dibujará el error de aproximación (con un gráfico contour) y discutirá su relación con la matriz hessiana.


[85: linmiso2]	Linealización de función de 2 variables; ejemplo Matlab, análisis error de linealización y Hessiano	***  09:20
Materiales: [Cód.: LinealizaMISOejemplo.mlx] [PDF]		[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo es continuación del vídeo [84]. Aquí se evalúa numéricamente en un cierto punto (incrementado desde el punto de operación) la exactitud de la linealización y, a continuación, se dibuja un mapa (contour) del error de aproximación.

Obviamente, cerca del origen, la diferencia entre la función original y la aproximación linealizada viene dominada por los términos de grado 2 del desarrollo de Taylor, por lo que las propiedades de la matriz hessiana de derivadas segundas permiten estimar el error de linealización y sus propiedades cerca del punto de operación: todos los autovalores del mismo signo indican contornos “elipsoidales” del error, autovalores de distinto signo indicarían contornos tipo “hipérbola” (no es el caso aquí).

3.2.1 Ecuaciones de sistemas dinámicos (linealización de EDO)

[86: lintdin]	Linealización (IV): sistemas dinámicos (ecuaciones algebraico-diferenciales)	**  10:34
Materiales: [lnteoriaMVDin.pdf]		[YouTube ►]

Este video aplica los resultados de linealización a sistemas dinámicos expresados como un sistema de ecuaciones diferenciales-algebraicas:

$$f\left(\frac{dx}{dt}, x, \zeta, u\right) = 0$$

siendo x denominado vector de *estado*, u el vector de *entradas* y z un vector de *variables adicionales* que pudieran intervenir en las ecuaciones (salidas, fuerzas de reacción, etc.).

El punto de linealización/tangencia elegido será un punto de equilibrio $\frac{dx}{dt} = 0$, $f(0, x_{eq}, \zeta_{eq}, u_{eq}) = 0$.

Como ejemplo, se propone linealizar un modelo de masa puntual deslizando sobre una curva con forma de parábola. Este vídeo se centra únicamente en la linealización, el detalle del modelado puede consultarse en los vídeos [60] y [61].

El ejemplo concluye obteniendo la representación interna normalizada. En esta representación se eliminan las variables ζ para obtener la ecuación de estado lineal normalizada (requiere que las ecuaciones estén planteadas de forma que determinada matriz jacobiana sea invertible, algunos sistemas de *índice superior* requieren pasos adicionales para obtener la representación interna, ver vídeo [30]). De hecho, la cinemática, discutida en el primero de los vídeos sobre la masa en una curva arriba referidos, es necesaria porque sólo con ecuaciones de posición los modelos mecánicos están bien planteados pero son de índice superior.

A veces, el concepto de dinámica lineal y superposición puede usarse de forma “intuitiva”, sin fórmulas, a partir de una serie de datos, como se discute en el vídeo [152].

3.3 Ejemplos adicionales

[87: lin1]

Simulación y linealización modelo calentamiento por radiación
(ejemplo Matlab ode45)

Materiales: [[CÓD.:](#) hornoej.mlx] [[PDF](#)]

**
27:01

[YouTube ▶]

Este video modela en forma de ecuaciones diferenciales (primer orden) un sistema donde un elemento a temperatura elevada calienta por radiación a otro elemento, que también intercambia calor por convección/conducción con el ambiente.

Se calcula un punto de equilibrio y se linealiza alrededor del mismo. Si estás empezando con esto, quizás te convenga visualizar antes los ejemplos de los vídeos [80] y [82], donde no intervienen conceptos de sistemas dinámicos.

Se simula con `ode45` el sistema no-lineal, y también el sistema dinámico linealizado. Se comprueba cómo ambas simulaciones se parecen si la temperatura de entrada está cercana al punto en el cual se ha ejecutado la linealización, pero que no es así si los incrementos de temperatura de entrada son elevadas.

Sistemas mecánicos

[88: mod2mass]

Modelado y linealización de un sistema no lineal de dos masas y dos muelles (Matlab)

Materiales: [[CÓD.:](#) modelado2muelles.mlx] [[PDF](#)]

**
08:19

[YouTube ▶]

Este vídeo modela y linealiza un sistema de dos masas, dos muelles y dos amortiguadores. Uno de los muelles es no lineal. Depende de tu familiaridad con estos conceptos, quizás podría ser aconsejable visualizar antes un caso lineal de varias masas y muelles en los vídeos [24] y [25].

Tras escribir un modelo (con seis señales intervinientes), se transformará a ecuación de estado y de salida (no lineal). Luego, se calculará un punto de funcionamiento con `solve` y se linealizará con `jacobian` y `subs`. Con ello, se tendrá un modelo lineal en representación interna para poder analizar sus propiedades en otros materiales.

[89: tiov11]

Dinámica lateral tiovivo (3): linealización (modelo no normalizado)

Materiales: [[CÓD.:](#) tiovivo1GLLinealiza.mlx] [[PDF](#)]

16:52

[YouTube ▶]

Este vídeo linealiza el modelo dinámico que se dedujo a partir de leyes elementales de primeros principios de la Física en el vídeo [52]. En concreto, hace una linealización “manual” en el punto de funcionamiento (estática, equilibrio) que también se usó como ejemplo en el referido vídeo. Aunque existían dos modelos no lineales (el original de 4 ecuaciones y la ecuación de estado normalizada de 2 ecuaciones), en este vídeo se linealiza el primero de ellos.

En el vídeo [90] se linealizará la ecuación normalizada y se comprobará que el resultado final es idéntico. En este vídeo y en el referido, se hace la linealización “a mano” con el objetio didáctico de aprender a hacerlo para un examen “lápiz y papel”... El comando `jacobian` ahorra *mucho* trabajo, como se discute en el vídeo [91] que sería la forma “rápida” recomendada de linealizar con Matlab.

También se simularán los modelos linealizados resultantes, en vídeos pendientes de publicar en este momento.

Una alternativa factible en algunos casos es linealizar con un ‘cambio de variable’ en entradas o salidas... SI sale ω^2 cambiar a $\nu = \omega^2$ y ya no hay no-linealidad. Esto se discute en el vídeo [93], pero NO suele ser lo que se explica y evalúa en primeros cursos sobre estos temas.

[90: tiov12] Dinámica lateral tiovivo (4): linealización a partir de ecuación de estado normalizada no lineal

Materiales: [CÓD.: tiovivo1GLLinealiza.mlx] [PDF]

*** 07:31

[YouTube ▶]

Este (breve) vídeo linealiza la ecuación en variables de estado, esto es, la representación interna normalizada $dx/dt = f(x, u)$ del movimiento lateral de un tiovivo modelado en el vídeo [52], obteniendo directamente una representación $dx/dt = Ax + Bu$ normalizada. Compara la linealización con la linealización resultante de primero linealizar y luego pasar a ecuación de estado normalizada, vista en el vídeo [89]. Como era de esperar, el resultado final es idéntico.

En este vídeo y en el arriba referido, se hace la linealización “a mano” con el objeto didáctico de aprender a hacerlo para un examen “lápiz y papel”... El comando jacobian ahorra *mucho* trabajo, como se discute en el vídeo [91] que sería la forma “rápida” de linealizar con Matlab delante.

[91: tiov13] Dinámica lateral tiovivo (5): linealización de ecuación de estado con “jacobian”, forma $Ax+Bu$

Materiales: [CÓD.: tiovivo1GLLinealizaShortYSimula.mlx] [PDF]

*** 10:00

[YouTube ▶]

Este video va “al grano” para modelar y linealizar el tiovivo no lineal objeto de los vídeos [51] (estática), [52] (dinámica), [89] (linealización “a mano” forma no normalizada) y [90] (linealización a mano en forma normalizada).

En vez de hacerlo de “varias formas” como en los otros vídeos (para entender bien la metodología y las diferentes alternativas), aquí se modela directamente con las dos ecuaciones de estado, se simula con ode45, se calcula punto de equilibrio con vpasolve, y se linealiza usando jacobian (matriz de derivadas parciales), con el mínimo código Matlab necesario (entendido “mínimo” informalmente).

Si la metodología está bien comprendida, el código de este vídeo es el que sería el recomendable, por su mayor concisión, frente a las versiones más largas con objetivos didácticos diferentes de los arriba referidos vídeos.

La parte final del vídeo muestra cómo introducir el modelo linealizado en forma matricial $dx/dt = Ax + Bu$, $y = Cx + Du$ en el “control systems toolbox” para, por ejemplo, obtener respuesta ante entrada constante (escalón, step) o senoidal (respuesta en frecuencia, bodemag).

[92: tiov14] Dinámica lateral tiovivo (6): comparativa modelo aproximado linealizado versus no lineal (ode45)

Materiales: [CÓD.: tiovivo1GLLinealizaShortYSimula.mlx] [PDF]

*** 10:57

[YouTube ▶]

En este vídeo se compara el modelo no lineal original de la dinámica de oscilaciones laterales de un tiovivo (modelo obtenido en vídeo [52]), y su linealización (Taylor primer orden), obtenida en el vídeo [91] (si se desea más detalle de todo el proceso, puede visualizarse el vídeo [90] antes).

Para ello, se simula el modelo no lineal hasta que alcanza un punto de equilibrio a 0.24 radianes de inclinación, y luego se baja un poquito (en forma de rampa) la velocidad de giro del armazón del tiovivo, con lo que se produce un pequeño cambio en el ángulo, cuya respuesta temporal se obtiene mediante ode45.

A continuación, se simula el modelo linealizado también con ode45 (aunque podría hacerse con lsim, ver vídeo [34] para detalles), y se superponen ambas simulaciones, tanto en coordenadas “incrementales” como en “absolutas”; como era de esperar, la respuesta temporal no lineal y la linealizada se parecen bastante.

Como contenido opcional, el vídeo [93] plantea la posibilidad de linealizar (parcialmente) con cambio de variable $\nu = \omega_0^2$; evidentemente, se puede hacer bajo ciertas suposiciones, y en dicho vídeo se reflexiona sobre esa idea.

[93: tiolv15] **Dinámica lateral ti vivo (7): linealización por cambio de variable (discusión conceptual)**

Materiales: [[CÓD.: ti vivo1GLLinealizaShortYSimula.mlx](#)] [[PDF](#)]

*** 07:49

[YouTube ►]

Este vídeo (opcional) discute sobre el caso de estudio del ti vivo cuya linealización se ha discutido en detalle en vídeos como [91] y [92], la idea de que como la entrada ω_0 aparece como ω_0^2 , podría ser posible aplicar el “truco” de cambiar $v = \omega_0^2$ de modo que así eliminaríamos un cuadrado si hacemos cálculos con la variable “artificial” v ... Esta idea, en general, se llamaría *linealización por cambio de variable*.

¿Es válida formalmente? Claro que sí; en este caso concreto cambiar una parábola por una recta introduce un error que si usáramos v como entrada no cometeríamos. No obstante, en el uso del modelo se requiere poder hacer cuadrados o raíces cuadradas... en determinadas aplicaciones de filtros o controladores diseñados con electrónica o electromecánica lineal, eso no es posible y por tanto la linealización a considerar sería la de $2\omega_{0,PF}\Delta\omega_0$.

En general, la presencia de no-linealidades invertibles a la entrada o salida de un proceso es fácil de “cancelar” con un cambio de variable. Si la no-linealidad está “dentro de la ecuación de estado”, entonces ya se hace más complejo, y hay que estudiar lo que se denomina *linealización por realimentación* que es la generalización de la idea que aquí se discute a sistemas dinámicos, de mayor dificultad conceptual, ver vídeos [502] o [505].

La linealización de un modelo de aeronave simplificado se aborda en el vídeo [fugeqlin(19:30)].

Procesos de calentamiento/mezclado en ingeniería química

[94: term1lin] **Tanque de calentamiento de líquido de primer orden: linealización y análisis de propiedades**

Materiales: [[CÓD.: HeatExch1elementLIN.mlx](#)] [[PDF](#)]

** 14:11

[YouTube ►]

Este vídeo revisa rápidamente el modelado de un tanque de calentamiento de primer orden y su simulación con [ode45](#). Si se desea una descripción más detallada de la obtención del modelo, se puede visualizar el vídeo [27]; asimismo, un análisis más detallado del comportamiento no lineal en simulación se aborda en el vídeo [33]. La visualización previa de estos vídeos es aconsejable.

Con propósito de linealizar, se selecciona un punto de funcionamiento de las entradas y se calcula el valor en equilibrio de la temperatura de salida.

A continuación, se obtienen las derivadas parciales del modelo respecto a los estados (temperatura) y a las entradas que permitirán escribir dicho modelo en forma normalizada lineal en representación interna $\dot{x} = A_{1 \times 1}x + B_{1 \times 3}u$.

El valor de A (como es de primer orden es una matriz 1×1) es el polo de la respuesta libre del sistema (con $u = 0$, la respuesta libre es $x(t) = e^{-At}x(0)$). Se analiza la rapidez de dicha respuesta en función del caudal (la inversa del polo es la constante de tiempo, muy parecida al tiempo de residencia si el tanque está bien aislado) y de la constante de disipación de calor al exterior.

La parte final compara la simulación del modelo linealizado (haciendo y deshaciendo los adecuados cambios a variables incrementales) con el modelo no lineal original. Se observa que si el caudal cambia mucho entonces la aproximación lineal no es buena, mientras que si los incrementos de caudal alrededor del valor nominal son pequeños, entonces el modelo linealizado describe muy bien la respuesta simulada.

La representación en función de transferencia de la linealización obtenida (así como para otros tipos de perfil de temperatura en el interior del tanque) se discute en el vídeo [183]. La función de transferencia exacta solución de las ecuaciones diferenciales en derivadas parciales de un calentador tubular se obtiene en el vídeo [184].

[95: linmix] **Linealización de un modelo tanque de mezclado, simulación comparada con original (ode45, Matlab)**

Materiales: [[CÓD.:](#) LinealizTanqueMezclado.mlx] [[PDF](#)]

** 10:55

[YouTube ▶]

En este vídeo se linealiza el modelo del tanque de mezclado cuyo modelo no lineal de segundo orden se desarrolló en el vídeo [22]. Los detalles sobre la simulación de este tanque con `ode45` se discuten en el vídeo [36]. Los contenidos de modelado y simulación serán referidos muy brevemente en este vídeo, se recomienda al lector visionar los vídeos referidos para más detalle.

En la ecuación de salida, ahora buscaremos simular/linealizar un modelo de la concentración de componente A , x_A y nivel del tanque h .

El primer paso para linealizar consiste en seleccionar un punto de funcionamiento (equilibrio). Se selecciona $q_{a,in} = 3$, $q_{b,in} = 1$. Fijando estas dos variables y las derivadas a cero, el comando `vpasolve` calcula los valores de todo el resto de variables (volúmenes acumulados, concentraciones, caudal de salida y nivel) para que se mantenga un equilibrio constante.

El comando `jacobian` calcula las derivadas parciales necesarias respecto a estados y entradas, tanto en ecuación de estado como de salida, para obtener una representación interna normalizada lineal.

La parte final del vídeo simula con `ode45` el modelo original y el modelo linealizado (éste último con entradas y condiciones iniciales incrementales). Deshaciendo el cambio de variable a incrementos de la solución linealizada se comparan ambas simulaciones, comprobando que el modelo linealizado tiene una exactitud razonable.

Como alternativa, que produce el mismo resultado, la linealización “automática” de un modelo Simulink de este tanque se aborda en el vídeo [97].

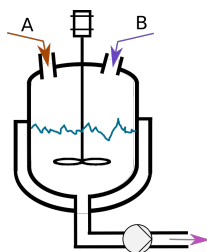
[96: linmixqs] **Modelado, Linealización y Simulación de tanque de mezclado con 3 entradas (Matlab)**

Materiales: [[CÓD.:](#) LinealizTanqueMezclaQS.mlx] [[PDF](#)]

*** 09:45

[YouTube ▶]

Este vídeo discute el modelado de un tanque de mezclado de dos componentes con extracción forzada, representado en la siguiente figura:



En este tanque, dos caudales $q_{a,in}$ y $q_{b,in}$ de componentes A y B son mezclados en el tanque. Intencionalmente, el sistema es *muy* similar al del vídeo [22]: la diferencia ahora es que el caudal de salida ya no depende del nivel, sino que es fijado de forma “exógena” por bombas aguas abajo, con lo que aunque sea una “salida de material”, desde el punto

de vista de teoría de sistemas, q_s es una variable de *entrada*. Por lo tanto, el modelo tiene una entrada *más* y, obviamente, una ecuación *menos* que el considerado en [22].

El modelado, representación interna y posterior linealización son paralelos a los desarrollos de los vídeos [22] y [95]. La diferencia es que en este sistema, para que exista equilibrio debe cumplirse que la suma de caudales de entrada y salida sea cero... entonces en equilibrio las tres entradas no son “independientes” y, por otro lado, si suman cero entonces cualquier nivel es un punto de equilibrio (equilibrio indiferente en Física, sistema marginalmente inestable en teoría de control). Estas cuestiones (existencia de infinitas soluciones en un sistema de ecuaciones) hacen que `solve` se pueda “liar” un poco... si sospechamos de esta situación, llamar a `solve` con la opción ‘ReturnConditions’ activada permite que el Symbolic Toolbox nos devuelva las infinitas soluciones de nivel que existen en este depósito, como se esperaba intuitivamente.

Una vez esa cuestión aclarada, todo es prácticamente igual que en los otros vídeos referidos, que abordaban el caso donde q_s era función del nivel h .

Nota: en muchas ocasiones, las entradas de “caudal” en procesos industriales lo son porque hay un sistema interno (control en cascada esclavo) de control de caudal, como el descrito en el vídeo [543], cuya visualización sería sólo aconsejable a estudiantes avanzados, dada su dificultad. ****

[97: slkmixLin]
Linealización de modelos Simulink: ejemplo tanque de mezclado

⌂
08:44

Materiales:
[CÓD.: LinealizTanqueSimulink.mlx] [PDF]
[YouTube ▶]

En este vídeo se demuestra cómo utilizar el comando `linearize` que linealiza un modelo de Simulink en los que se han marcado determinadas señales como *Linear Analysis Points*. Estos puntos son extraídos con el comando `getlinio`, y los tiempos donde extraer el punto de operación son argumentos a `linearize`.

El vídeo compara la linealización obtenida con la linealización obtenida mediante `jacobian` de la Symbolic Toolbox, y se ve que coinciden. Por hacerlo autocontenido, los minutos [01:30] a [03:30] describen brevemente dicha linealización por jacobiano, cuyos detalles se presentan en el vídeo [95].

El vídeo también comprueba que la respuesta ante escalón de dos modelos linealizados en puntos diferentes exhibe grandes diferencias. Aunque el análisis en detalle queda fuera de los objetivos de este vídeo, ello puede causar problemas a los sistemas de control lineales en este tipo de sistemas, necesitando estrategias que aseguren tolerancia a error de modelado (control robusto) o de planificación de ganancia (cambiar controlador según el punto de funcionamiento).

Nota: En la Sección 29.3 se generaliza la linealización de un sistema alrededor de un equilibrio constante al caso de linealizar alrededor de una *trayectoria de referencia* de entradas y salidas. El resultado es un sistema lineal *variante en el tiempo*, cuyo análisis, obviamente, reviste mayor complejidad. Esta es la razón por la que dicha linealización ha sido considerada en los capítulos finales que abordan temas más complejos. ⌂

3.4 Autoevaluación

1. ¿Cuál es la motivación principal para linealizar una función no lineal?
2. ¿Qué condición debe cumplir una función para ser considerada lineal formalmente?
3. ¿Cómo se representa geométricamente una función lineal de una variable?

4. ¿Qué concepto geométrico se utiliza para aproximar una función no lineal mediante una función lineal en la linealización?
5. ¿Qué representa la derivada de una función en un punto dado en el contexto de la linealización?
6. En la ecuación de la recta tangente $(y - y_0) = m(x - x_0)$, ¿qué representa la variable "m"?
7. ¿Cómo se expresa la linealización de una función en variables incrementales?
8. ¿De qué depende la elección del punto de linealización en la práctica?
9. ¿Cómo se calcula la pendiente de la recta tangente en un punto dado de una función?
10. ¿Qué representa la serie de Taylor en el contexto de la linealización?
11. ¿Qué es el método de Newton y cómo se relaciona con la linealización?
12. ¿Qué es la matriz jacobiana y cómo se utiliza en la linealización de funciones multivariantes?
13. ¿Qué significa trabajar con "coordenadas incrementales" al linealizar una función?
14. ¿Qué es un punto de equilibrio en un sistema dinámico?
15. ¿Cuál es la importancia del punto de equilibrio al linealizar un sistema dinámico?
16. Explica el concepto de "representación interna normalizada" en un modelo linealizado y su expresión.
17. ¿Por qué es importante no confundir linealización con otros conceptos como regresión lineal o acotación lineal?

Capítulo 4

Respuesta temporal

La respuesta temporal de un sistema físico ante unas ciertas entradas requiere resolver la ecuación diferencial que constituye el modelo matemático de dicho sistema.

Algunos tipos de ecuaciones diferenciales (en concreto, las *lineales*) pueden ser resueltas explícitamente (obteniendo una fórmula en función del tiempo para la solución) y otras no, y sólo pueden ser resueltas numéricamente (integración numérica, simulación, sección 2.1).

El siguiente vídeo ilustra el uso del comando `dsolve`, de la *Symbolic toolbox* de Matlab, para obtener la solución explícita de algunos tipos de ecuaciones diferenciales.

► *Resolución de ecuaciones diferenciales con Matlab* :

[Fernando Giménez]

[Enlace]

4.1 Transformada de Laplace

El primer método que se suele explicar en cursos de grado para el cálculo de la respuesta temporal se basa en la Transformada de Laplace.

4.1.1 Función de transferencia (SISO), Matriz de transferencia (MIMO)

[98: fdt] La función de transferencia

controltheoryorg (canal YouTube)

** 16:03
[PLAY ►]

Este vídeo introduce la transformada de Laplace de una ecuación diferencial

$$y^{(n)} + a_{n-1}y^{(n-1)} + \dots + a_1y' + a_0y = b_my^{(m)} + b_{m-1}u^{(m-1)} + \dots + b_1u' + b_0u$$

donde aparece un término de condiciones iniciales.

Después, suponiendo al sistema dinámico inicialmente en reposo (condiciones iniciales de entrada y salida igual a 0, punto de funcionamiento-equilibrio), obtiene una transformada de Laplace mucho más sencilla: $Y(s) \cdot (s^n + a_{n-1}s^{n-1} + \dots) = U(s) \cdot (b_ms^m + b_{m-1}s^{m-1} + \dots)$, que puede expresarse como:

$$\frac{Y(s)}{U(s)} = \frac{b_ms^m + b_{m-1}s^{m-1} + \dots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0} := G(s)$$

denominándose a $G(s)$ como *función de transferencia*.

Define los polos y ceros como las raíces de denominador y numerador, respectivamente, el concepto de causalidad ($n \geq m$) y esboza la posibilidad de la identificación experimental de la función de transferencia de un sistema a partir de entradas y salidas sin conocer su “física interna”.

Los modelos que incorporan varias entradas y salidas (multivariables) deben ser descritos con la generalización del concepto de función de transferencia al de *matriz* de transferencia, vídeo [131].

Ejemplos numéricos de su uso para cálculo de respuesta temporal se pueden consultar en los vídeos [111], [112], [113], [114].

Videos de terceros relacionados:

Aquí se enlazan algunos materiales de terceros relacionados, utilizando la Transformada de Laplace:

► *Cálculo de la antitransformada de Laplace por descomposición en fracciones simples:* [Macarena Boix]

[Enlace]

► *Transformada de Laplace con Mathematica:*

[Elena Sánchez]

[Enlace]

► *Aplicación de la Transformada de Laplace a la resolución de problemas de valores iniciales:* [P. Car-men]

[Coll]

[Enlace]

4.2 Respuesta temporal de sistemas sencillos

[99: stdinp]

Entradas estándar para el análisis de sistemas (escalón, rampa, senoidal): definición y utilidad

controltheoryorg (canal YouTube)

**



11:58

[PLAY ►]

Este vídeo plantea el problema de cómo interpretar los polos/ceros/ganancia de una función de transferencia a partir de la respuesta ante entradas típicas: aplicando las mismas entradas “típicas” a todos los sistemas nos permite interpretar y comparar sus características, así como comunicar características del funcionamiento a técnicos ajenos al control en términos “temporales”.

Presenta las entradas impulso, escalón, rampa, parábola, y sinusoidal (cuando la frecuencia varía se denomina respuesta en frecuencia).

4.2.1 Caso de estudio masa-muelle-amortiguador

[100: mmamexp]

masa resorte amortiguador real

Señales y Sistemas (canal YouTube)

*



03:00

[PLAY ►]

Este vídeo presenta un sistema experimental de masa-muelle-amortiguador como los ampliamente usados en los libros de texto de dinámica de sistemas. Un vistazo a cosas reales en vez de sólo a ecuaciones viene bien de tanto en tanto, y es muy corto.

[101: masmusym1]

Resolución ecuaciones diferenciales ordinarias (EDO) con Matlab (dsolve): respuesta libre masa-muelle

Materiales: [Cód.: testEDOsym.mlx] [PDF]

**



11:30

[YouTube ►]

* LINK TO ENGLISH VERSION

Este vídeo plantea la ecuación diferencial de un sistema masa-muelle con rozamiento $M \frac{d^2y}{dt^2} = -ky - b \frac{dy}{dt}$ y, sustituyendo los parámetros constantes por sus valores numéricos obtiene la solución general (con dos constantes de integración) mediante `dsolve`. Dicha solución es analizada pero para comprender mejor la EDO se resuelve, también con el mismo comando, dando valores de posición y velocidad iniciales. La solución se representa gráficamente y se anima para entender el tipo de movimiento obtenido (oscilaciones amortiguadas).

El comando `dsolve` permite obtener una expresión explícita de la fórmula de la respuesta, en vez de sólo un conjunto de puntos como las simulaciones numéricas `ode45`; el mismo sistema masa-muelle se aborda en el vídeo [32] y la comparación `ode45` versus `dsolve` se aborda en el vídeo [105]. La resolución por transformada de Laplace se aborda en el vídeo [107].

Simulaciones adicionales variando parámetros y el código Matlab de animación se detalla en el vídeo [102]. La EDO en representación interna se aborda en el vídeo [103], obteniendo resultado idéntico, como era de esperar,

[102: masmuAnim]
10:36

Animación respuesta temporal masa-muelle en Matlab, efecto del amortiguamiento

Materiales: [[Cód.:](#) simulspringdamper.m] [[PDF](#)]

[\[YouTube ▶\]](#)

* [LINK TO ENGLISH VERSION](#)

Este vídeo presenta resultados de animación de la respuesta libre de un sistema masa-muelle (con amortiguación por rozamiento) obtenidos en los vídeos [101] y [103] mediante `dsolve`. Los comandos básicos son `patch`, `xline`, `yline`, `plot`, dentro de un bucle que genera cada “frame” (no en tiempo real, esto no es un videojuego).

Realmente, el detalle del código es meramente “cosmético” con funciones de dibujado, no se aborda ningún tema importante desde el punto de vista conceptual sobre resolución de ecuaciones diferenciales.

También se comprueba lo que ocurre al disminuir y aumentar el coeficiente de fricción.

Un ejemplo de código, simular pero más laborioso, para animación de las vibraciones de un sistema de tres masas y cuatro muelles es descrito en el vídeo [35].

[103: masmusym2]

Resolución ecuaciones diferenciales ordinarias (EDO) con Matlab (dsolve): respuesta libre masa-muelle representación interna

** 07:53

Materiales: [[Cód.:](#) testEDOsym.mlx] [[PDF](#)]

[\[YouTube ▶\]](#)

* [LINK TO ENGLISH VERSION](#)

Este vídeo obtiene la respuesta temporal libre ante condiciones iniciales fuera de equilibrio de un sistema masa-muelle, abordando exactamente el mismo problema de segundo orden que el vídeo [101] $M\ddot{y} = -ky - b\dot{y}$. En este vídeo, sin embargo, se aborda planteando las ecuaciones en representación interna normalizada $\frac{dx}{dt} = Ax$, con $A = \begin{bmatrix} 0 & 1 \\ -k/M & -b/M \end{bmatrix}$. El comando `dsolve` no tiene ningún problema con dicha representación y obtiene, obviamente, resultados idénticos al del vídeo referido. El código de animación se detalla en el vídeo [102]. La resolución por transformada de Laplace se aborda en el vídeo [107].

El uso de `dsolve` es ilustrado en el vídeo [120] para sistemas con *retardo*.

[104: masmusymForz]
12:59

Resolución EDO con Matlab (dsolve): respuesta forzada masa-muelle (constante + senoidal)

Materiales: [[Cód.:](#) testEDOsymForzada.mlx] [[PDF](#)]

[\[YouTube ▶\]](#)

* [LINK TO ENGLISH VERSION](#)

Este vídeo presenta la solución de la EDO de un muelle $M\ddot{y} = -ky - b\dot{y} + F(t)$ cuando es forzado con una fuerza con componentes constantes y senoidales.

Para un análisis más pausado de sólo la respuesta libre, visiona el vídeo [101].

Se comprueba que aparecen términos "parecidos" a la entrada (constantes indicando el nuevo equilibrio si la fuerza tiene un componente constante, y vibraciones de la misma frecuencia que la entrada (régimen estacionario senoidal), así como términos con componentes de la respuesta libre, con tasa de decaimiento y frecuencia propia de oscilaciones que no depende de las características de la entrada $F(t)$.

El resultado se grafica y se anima, usando el mismo código que el explicado en el vídeo [102].

Una respuesta forzada ante entrada tipo "rampa" se discute en el vídeo [105], así como la simulación forzada en forma de ecuación de estado normalizada $\dot{x} = f(t, x)$ de este muelle.

[105: dsolvevsode45] Resolución simbólica (dsolve) versus numérica (ode45): masa muelle amortiguador, ventajas e inconvenientes

14:10

Materiales: [[Cód.:](#) DsolveversusODE45ES.mlx] [[PDF](#)]

[YouTube ▶]

Este vídeo compara la resolución de una ecuación diferencial de segundo orden (muelle forzado con entrada tipo "rampa") mediante **dsolve**, que produce una fórmula función del tiempo, versus **ode45**, integración numérica (simulación) cuyo resultado es un conjunto de instantes y de estados.

Plantea el código necesario para generar las EDO simbólicas y la representación en variables de estado necesarias para **ode45**, y comprueba que las simulaciones son, como era de esperar, muy parecidas.

Un estudio más concienzudo de la parte "simbólica" se aborda en los vídeos [101] y [103], aquí revisado de forma rápida. También, un estudio más en detalle de la simulación numérica del muelle se aborda en el vídeo [32].

La parte final analiza las ventajas e inconvenientes de dsolve, ode45 e introduce **lsim** (simulación de sistemas lineales de la Control System Toolbox), como "intermedio". La comparación ode45 vs lsim en un sistema de 4 muelles se discute en el vídeo [34].

Por supuesto, para la resolución simbólica será necesaria teoría de ecuaciones diferenciales lineales, por ejemplo el método de Transformada de Laplace, discutido en el vídeo [107] y muchos otros ejercicios/ejemplos en la colección.

[106: sssmlk] Sistema masa-muelle-amortiguador: simulación en Simulink, extracción de representación interna y FdT

Eduardo Giraldo

*** 11:46

[PLAY ▶]


Este vídeo codifica en **Simulink** las ecuaciones de un sistema masa-muelle-amortiguador $\dot{y} = v$, $\dot{v} = -B/M \cdot v - K/M \cdot y + 1/M \cdot u$, mediante los bloques suma, ganancia e integrador de la librería de Simulink. Luego encapsula las ecuaciones en un subsistema (*create subsystem*) y crea una máscara donde se genera un cuadro de diálogo para introducir los valores de los parámetros constantes M , K y B . Con ello, obtiene la simulación ante un escalón.

En la fase final del sistema utiliza la **linear analysis tool** para obtener función de transferencia o modelos de espacio de estados linealizados del subsistema (como el sistema es lineal, la linealización que ejecuta Simulink devuelve el mismo sistema), y exportarlos al espacio de trabajo de Matlab para un análisis de propiedades o diseño de control posterior. Para comprobar la coincidencia de las respuestas compara la respuesta del subsistema con las de los bloques *state-space* y *Transfer Fcn* de la librería de Simulink.

Un caso con 3 masas y 4 muelles se aborda en el vídeo [24].

[107: masmuLapl]

Resolución EDO por transformada de Laplace: ejemplo
masa-muelle ante escalón
Materiales: [Cód.: testEDOsymLaplace.mlx] [PDF]

***  17:59

[YouTube ▶]

Este vídeo presenta la resolución por Transformada de Laplace de la respuesta forzada ante entrada constante de un sistema masa-muelle (con rozamiento). Dado que es el mismo problema que el tratado en el vídeo [104], los tres primeros minutos resumen las ideas principales de dicho vídeo y relacionados, que utilizan `dsolve` para resolver la EDO en una línea de Matlab (recomendado).

Posteriormente, se hace la transformada de Laplace de la ecuación diferencial, la transformada de Laplace de la entrada escalón, sustitución de condiciones iniciales y despeje de la salida (en dominio de Laplace, $Y(s)$).

Estas operaciones se hacen tanto manualmente como con los comandos `laplace` y `solve`.


La respuesta temporal se obtiene con la transformada inversa de Laplace (que Matlab realiza con `ilaplace`). Alternativamente, puede descomponerse $Y(s)$ en fracciones simples y consultar tablas de transformadas. Esa opción también se aborda, usando `partfrac` para la descomposición.

Todas las formas de obtener la solución (se revisan tres) dan el mismo resultado, obviamente.

4.2.2 Sistemas genéricos

[108: smktf]

Simulink: simulación de sistemas dinámicos en función de
transferencia
Juan M. Herrero Durá (UPV)


**  06:27

[PLAY ▶]

Este vídeo simula un depósito con dos entradas (caudal de perturbación, válvula de salida), linealizado alrededor de un punto de funcionamiento, que produce una representación $h(s) = G_1(s)u(s) + G_2(s)q_e(s)$. Utiliza los bloques *Step*, *Sum*, *Mux*, *Transfer Fcn* y *Scope*.

[109: stepord1]

Respuesta ante escalón unitario de sistemas de primer orden
controltheoryorg (canal YouTube)

**  14:16

[PLAY ▶]

Este vídeo describe un sistema de primer orden con un polo en $-p$, con $G(s) = K/(\frac{s}{p} + 1)$. Una explicación alternativa (pero básicamente similar) realizada por mí se presenta en el vídeo [136], con un ejemplo Matlab.

Aquí, se calcula la respuesta ante escalón unitario mediante la descomposición en fracciones simples de $G(s) \cdot \frac{1}{s}$, resultando en $y(t) = K \cdot (1 - e^{-pt})$. Se interpreta K como el valor *final* de la respuesta ante escalón unitario, p como la pendiente en $t = 0$. Define la constante de tiempo como $1/p$, y el sistema alcanza un 63% del valor final en τ . Despeja, por ejemplo, el *tiempo de establecimiento* t_{est} cuando resta el 1% de la respuesta (se ha completado el 99% del transitorio), $t_{est,0.01} = 4.6/p = 4.6\tau$,


Nota: Aunque no se discuten en detalle en el vídeo, es más común utilizar el tiempo de establecimiento 5%, resultado de resolver $(1 - e^{-pt_{est}}) = 0.95$, que resulta en aproximadamente $t_{est,0.05} = 3/p = 3\tau$, o el tiempo de establecimiento 2%, resultado de resolver $(1 - e^{-pt_{est}}) = 0.98$, que es igual a $t_{est,0.02} = 4\tau$.

Modificaciones sencillas cuando hay un “retardo” inicial, y el concepto de identificación experimental de este tipo de sistemas son abordadas en el vídeo [154], cuya visualización se aconseja.

Una interpretación/aplicación ‘intuitiva’ del significado de la respuesta de sistemas de primer orden se aborda en el vídeo [152].

[110: ejord1] Ejemplos de sistemas de primer orden (experimentales)

Antonio Barrientos (UPM)

*  06:09

[PLAY ►]


Este video perteneciente a la colección de Dinámica de SIstemas de Antonio Barrientos, presenta la respuesta *experimental* ante escalón de tres sistemas de primer orden: circuito resistencia-condensador, calentamiento de una sonda, velocidad de un motor eléctrico. Comprueba la similitud entre todas las respuestas experimentales y con las predicciones teóricas basadas en la función de transferencia, lo que justifica la gran importancia del análisis unificado del modelo $a/(s+b)$ o, equivalentemente, de $K/(\tau s+1)$ en las asignaturas de grado de dinámica de sistemas y control automático, discutida en el vídeo [109].

Obviamente, las escalas de tiempo son muy diferentes (minutos en sistema térmico, décimas de segundo en sistema mecánico, milisegundos en sistema eléctrico), pero aparte de la escala de tiempos, en el eje “vertical” todo se parece mucho.

De hecho, muchas veces se identifica un modelo del sistema directamente a partir de los datos de experimentos (lo que se denomina *identificación experimental*) en vez de a partir de las ecuaciones de la física. El caso de identificación experimental de sistemas de primer orden (+ retardo) es muy sencillo, y se aborda en el vídeo [154].

[111: ilaplaceex1] Respuesta temporal mediante fracciones simples y tablas de transformada inversa: sistemas 1er orden

Materiales: [[CÓD.:](#) PartFracDemoES.mlx] [[PDF](#)]

**  13:53

[YouTube ►]

Este vídeo presenta el cálculo de respuesta temporal de dos sistemas de primer orden, uno estable y con condiciones iniciales nulas, otro inestable y con condiciones iniciales no cero. La entrada en ambos es un escalón unitario.


Se comprueba la coincidencia del resultado de `dsolve` con el método de Laplace (descomposición en fracciones simples `partfrac`, transformada inversa `ilaplace`).

Los ejemplos del archivo MLX que acompaña a este vídeo continúan en el vídeo [112], [113] y [114].

Por supuesto, en aplicaciones también puede ser una opción la resolución numérica, vídeo [105], que incluso aplica a sistemas no lineales pero, por contra, no provee de “fórmula”.

[112: ilaplaceex2] Respuesta temporal mediante fracciones simples y tablas de transformada inversa: sistemas 2o orden polos reales

Materiales: [[CÓD.:](#) PartFracDemoES.mlx] [[PDF](#)]

**  15:58

[YouTube ►]

Este vídeo presenta el cálculo de respuesta temporal de dos sistemas de segundo orden por método de Laplace, uno con polos simples (función de transferencia $G(s) = 5/(s+1)/(s+2)$) ante escalón y otro con polos múltiples ante rampa ($u(t) = t$, $U(s) = 1/s^2$), con condiciones iniciales nulas.

El vídeo es análogo al vídeo [111] donde se presentaban ejemplos de primer orden más sencillos; se usan `dsolve`, `partfrac`, `ilaplace`.

Nota: El comando `partfrac` por defecto busca polinomios con coeficientes *racionales*; si el ejemplo está “preparado” (como es aquí el caso) para que las raíces sean “bonitas” (enteras o racionales), entonces `partfrac` funciona correctamente. En un caso genérico, se debería añadir como opción `FactorMode='real'` para decirle que busque fracciones simples con denominadores con raíces *reales*. Ver documentación de Matlab o/y vídeo [175] para ejemplo.

Los ejemplos del archivo MLX que acompaña a este vídeo continúan en el vídeo [113] y [114]. Más ejemplos de respuesta temporal con Laplace en vídeo [107] (masa-muelle polos complejos).

En algunos casos, existen retardos de transporte en los sistemas o en las entradas, que requieren ciertas modificaciones al proceso de cálculo de la respuesta temporal, según se detalla en los vídeos [120], [121] o [129].

[113: ilaplaceex3]

Respuesta temporal mediante fracciones simples y tablas de transformada inversa: sistemas 2o orden polos complejos

Materiales: [[CÓD.:](#) PartFracDemoES.mlx] [[PDF](#)]

*** 14:57
[YouTube ▶]

Este video continua explorando los conceptos de resolución de ecuaciones diferenciales lineales por transformada inversa de Laplace por fracciones simples + tablas, iniciado en los vídeos [111] y [112] en ejemplos con polos reales.

Aquí se aborda el cálculo de la respuesta ante escalón de un sistema de segundo orden oscilatorio (subamortiguado, polos complejos) mediante transformada de Laplace. El enfoque es muy similar al ejemplo del vídeo [107] donde se abordan las oscilaciones de un muelle (que da lugar al mismo tipo de ecuación diferencial que el aquí resuelto).

Los ejemplos del archivo MLX que acompaña a este vídeo continúan en el vídeo [114].

[114: ilaplaceex4]

Respuesta temporal mediante fracciones simples y tablas de transformada inversa: sistema 4o orden ante rampa

Materiales: [[CÓD.:](#) PartFracDemoES.mlx] [[PDF](#)]

**** 12:39
[YouTube ▶]

Este vídeo desarrolla un ejemplo de respuesta temporal por método de Laplace un poco más complicado que los vistos en los vídeos [111], [112] y [113], cuya visualización previa se recomienda.

En este caso, se plantea la respuesta ante rampa $u(t) = t$ de un sistema de orden 4 con un par de polos complejos y un polo doble en $s = -1$. Se compara el desarrollo cuasi-manual con `collect`, `simplifyFraction` y `solve` con el resultado obtenido directamente sin pasos intermedios por `ilaplace`, siendo coincidentes, como era de esperar.

Las respuestas incorporando retardo se ven en casos de estudio como [121] (primer orden con retardo), [130] (retardos ante rampa saturada) o [175] (motor, tren de escalones).

4.2.3 Otros ejemplos:

[115: pend1EN]

Unstable inverted pendulum: modelling, linearization, free response (Laplace)

Materiales: [pendumodel.pdf]

** 13:20
[YouTube ▶]

This video presents a pendulum model using Newton's equations and torque balances.

As the goal is analysing the behaviour close to its inverted (unstable) equilibrium position, $\sin \alpha \approx \alpha$ is then linearized in said upper equilibrium.

A Laplace transform of the pendulum ODE is then performed, obtaining the transfer function representation of the system (the input is a manipulable torque at the lower joint) + initial conditions term (dependent on the initial position and velocity at the start of the experiment/simulation).

This representation in the Laplace domain of the linearized model is used to calculate the free response from non-zero initial conditions and zero input, verifying that it, obviously, tends to infinity (unstable response). This does not occur in the real nonlinear system, but the linearized model is only locally valid for small angular deviations around the upper vertical position.

[116: rcstepsin]

Circuito RC serie: modelado y respuesta entrada
escalón/senoidal por transformada de Laplace

***  18:48

Materiales: [[CÓD.:](#) RCRespuestaSenoide.mlx] [[PDF](#)]

[YouTube ►]

En este vídeo se presenta cómo calcular la respuesta temporal de un circuito RC serie. Se desea la respuesta tanto de la tensión en condensador como de la intensidad. El manejar dos funciones de transferencia a la vez formalmente se considera trabajar con una *matriz de transferencia* (definición y detalles en vídeo [131]), pero (al menos en este caso concreto) no deja de ser más que cuestión de notación, y podrían haberse abordado dos problemas “separados” de Laplace.

El hecho es que primero las respuestas ante escalón (corriente continua) y senoide (corriente alterna) se calculan en una línea con los comandos **step** y **bode** de la Control System Toolbox (si estás empezando, no te preocupes mucho del diagrama de Bode en este momento, no es el objetivo principal).

El objetivo principal es ilustrar la metodología de Laplace, por lo que se multiplica la función (matriz) de transferencia por la entrada ($1/s$ en el caso escalón, $\omega/(s^2 + \omega^2)$ en el caso senoidal), se hace descomposición en fracciones simples mediante **partfrac**, y se obtiene la transformada inversa (manualmente, mirando tablas a partir de las fracciones simples; con Matlab, directamente con **ilaplace**). Se analiza la respuesta, identificándose un transitorio exponencial e^{-250t} que aparece en todas las respuestas, junto a un régimen permanente constante en el caso escalón, y senoidal en el caso de corriente alterna. Se analiza el concepto de “desfase” en ese régimen permanente senoidal.

Opcionalmente, la parte final del vídeo discute la comparación del régimen permanente senoidal obtenido mediante Laplace con el obtenido con fórmulas de impedancia usuales en cursos de ingeniería eléctrica, o con la fórmula de respuesta en frecuencia de un sistema genérico. Obviamente coinciden. Como se ha dicho, esta última parte es opcional si ya conoces esas ideas antes de visualizar este vídeo; si no, concéntrate en la parte central (Laplace).

[117: LinrushlinEN]

Inrush current when energising an inductor, linear ODE case
study

*** 

22:52

Materiales: [[CÓD.:](#) LinearInductorInrush.mlx] [[PDF](#)]

[YouTube ►]

In this video, the transient overcurrents (inrush current) in an inductor (say, a transformer) are studied as an example of applications of ODE solving to electrical engineering. Only the *linear* inductor case is studied in this first video, leaving magnetic saturation for forthcoming videos.

First, the circuit model is stated $\frac{di}{dt} = \frac{1}{L}(V - Ri)$, being V the input voltage to a series resistor-inductor circuit.

Then, the ODE is solved for initial condition $i(0) = 0$. The result contains an exponential transient which adds to a steady-state solution which is a sinusoidal waveform (steady-state AC current).


Plotting the transient, we discover that initial current peak is twice the amplitude of the steady-state current.

The initial phases of the response are close to that of a pure inductor. The final phases of the response approach, when $t \rightarrow \infty$, i.e., the solution computed with impedances $Z(\omega) = R + L\omega j$, results in $I(\omega) = \frac{1}{Z(\omega)} \cdot V(\omega)$.

The last minutes of the video discuss how to avoid the inrush overcurrent phenomenon: timing the transformer connection so that suitable power electronics equipment connects the unloaded inductor when input voltage has the peak value (approximately). Repeating the plots of the new solution, initial conditions do match the steady-state waveforms so exponential transient and the overcurrent associated to it are eliminated.

Inrush overcurrents are exacerbated if the inductor core undergoes magnetic saturation; these issues are out of the scope of the present video, for brevity and simplicity.

Python version: <https://personales.upv.es/asala/DocenciaOnline/Material/Python/linind2.py>

[118: fugeqlin] **Dinámica fugoide aeronave: equilibrio, linealización, estabilidad**
***  19:30


Materiales: [[CÓD.](#): FugoideLinealizEstabES.zip] [[PDF](#)]
[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute el cálculo analítico de puntos de equilibrio, su linealización y el análisis de estabilidad, de los modelos de dinámica fugoide simplificada (orden 2) de una aeronave. El detalle de la obtención del modelo basado en principios de la Física se discute en el vídeo [57], y diferentes simulaciones con `ode45` y animaciones se abordan en el vídeo [58].

En este vídeo, se discute sobre las relaciones entre el ángulo, la velocidad aerodinámica y el empuje necesarios en equilibrio, así como la estabilidad de dicho equilibrio. Se hace énfasis en el vuelo sin motor (planeador $u = 0$) y en el vuelo nivelado horizontal ($\theta = 0$) como casos particulares. La estabilidad se analiza obteniendo una representación en variables de estado normalizada $\dot{x} = Ax$ para empuje constante, y evaluando la parte real de los valores propios de A , soluciones de la ecuación característica $\det(sI - A) = 0$. Por supuesto, como en todo sistema linealizado, la estabilidad del mismo sólo prueba la estabilidad “local” del sistema no-lineal original.

4.2.4 Sistemas con retardo en dinámica, o entradas “por tramos”

[119: retard] **Sistemas con retardo: introducción y aproximación**
**  13:17

Antonio Barrientos (UPM)
[PLAY ►]

Este vídeo presenta el concepto de retardo: el término e^{-Ts} en Transformada de Laplace y cómo afecta la respuesta de un sistema cuando está en el “numerador” de una FdT. Presenta el problema del retardo interno/realimentado, cuyo efecto ya no es tan fácil de interpretar.

Presenta ejemplos de sistemas industriales con retardo: tren de laminación con sensor desplazado, medida de concentración en un fluido aguas abajo de un mezclador, tiempo

de incubación en sistema biológico, tiempo de análisis (cromatógrafo), retardo de comunicación/teleoperación, etc.

La parte final discute la idea de aproximar un retardo por un polinomio (serie de Taylor truncada), y la aproximación de Padé (que resulta en un cociente de polinomios), para aplicarlo en aquellas metodologías de análisis o diseño de controladores que necesiten un sistema de orden finito.

El retardo de transporte ocurre en muchos fenómenos físicos, e interviene en la solución de muchas ecuaciones en derivadas parciales (ver vídeo [184], por ejemplo, para ver cómo aparece en la solución a un intercambiador de calor tubular). Obviamente, no es necesario ser un experto en ecuaciones en derivadas parciales para entender conceptualmente el significado del retardo en control de procesos.

La obtención analítica de la respuesta temporal de sistemas de primer orden con retardo es discutida en los vídeos [120] y [121]. Aproximaciones del retardo son discutidas en el vídeo [191].

Sistema de primer orden con retardo

[120: dly1er1]

Respuesta temporal sistema primer orden con retardo de transporte: `dsolve`, discusión sobre resultados

Materiales: [[CÓD.:](#) PartFracDemoESDelay.mlx] [[PDF](#)]

**
05:57

[YouTube ▶]

Este vídeo obtiene la respuesta temporal ante entrada escalón y condiciones iniciales nulas de un sistema de primer orden con retardo de transporte $\frac{dy}{dt} + 3y(t) = u(t-2)$, mediante el comando `dsolve` de la Symbolic Math Toolbox de Matlab. El detalle de cómo obtener dicha respuesta (por ejemplo mediante Transformada de Laplace) será objeto de otro vídeos continuación de este, en concreto [121], de próxima publicación.

*Otros ejemplos de uso de `dsolve` en sistemas sin retardo aparecen en los vídeos [101], [103].

El objetivo aquí es motivar por qué aparece esta dinámica, poniendo un ejemplo de un tanque de calentamiento cuya alimentación sea a través de una tubería “larga” de modo que haya un fenómeno de transporte que dé lugar a ese retardo (si tienes interés en el detalle, el modelado del tanque de calentamiento sin retardo se aborda en el vídeo [27], aunque no es necesario para comprender lo aquí explicado). Ejemplos adicionales de motivación hacia el estudio de los sistemas con retardo pueden verse en el vídeo [119].

A continuación, se presenta cómo resolver con `dsolve` la EDO con retardo asociada. Se necesita el comando `heaviside` para indicar funciones a trozos que valen cero hasta un cierto instante (no es lo mismo la constante “2” que el “escalón de 2 unidades en $t = 0$ ”; la segunda de las funciones debe codificarse en el Symbolic Toolbox de Matlab con `2*heaviside(t)`).

La parte final del vídeo discute la interpretación “intuitiva” de la solución obtenida (igual a la de un sistema de primer orden, pero retrasada).

Este tipo de modelos es muy habitual en ingeniería de procesos; la identificación experimental de los mismos se aborda en el vídeo [154].

Realmente el retardo es la solución de la “ecuación en derivadas parciales (EDP) de transporte”, aunque si estás empezando en temas de automática, seguramente no tendrás interés en complicarte la vida con EDP; si lo tienes, puedes ver el vídeo [191].

[121: dly1er2]

Respuesta temporal sistema primer orden con retardo: resolución por transformada de Laplace

Materiales: [[CÓD.:](#) PartFracDemoESDelay.mlx] [[PDF](#)]

12:29

[YouTube ▶]

Este vídeo obtiene, por métodos de transformada de Laplace, la respuesta ante escalón del sistema de primer orden con retardo $\frac{dy}{dt} + 3y(t) = u(t - 2)$. La respuesta con `dsolve` ya se abordó en el vídeo [120], así como un análisis “intuitivo” de por qué aparecen retardos y qué características tiene la respuesta. Si estás empezando en estos temas, te aconsejo ver el referido vídeo antes que éste.

La solución se aborda mediante dos métodos (equivalentes):

Primero, resolver la EDO sin retardo (por Laplace o por el método que queramos, realmente) y utilizar la idea de *invarianza temporal* de modo que podemos decir que la salida ante un retardo a la entrada de 2 unidades de tiempo será idéntica a la salida sin retardo en la entrada, excepto que se producirá 2 unidades de tiempo después. Por tanto, primero se hace una resolución del estilo de la hecha en el vídeo [111], y luego se retrasa “manualmente” para incorporar la idea del retardo. Para representarlo gráficamente, se usa la Symbolic Toolbox, incorporando el comando `heaviside` en la descripción de la solución. El resultado, obviamente, coincide con el obtenido por `dsolve` en el primer vídeo arriba referido.

La segunda opción es incluir el retardo en las manipulaciones en el dominio de Laplace, utilizando el hecho de que la transformada de Laplace del operador retardo es e^{-ds} siendo d el tiempo de retardo. Al hacerlo, hay que manejar dicha exponencial de la variable de Laplace en la descomposición en fracciones simples hasta poder llegar a la transformada inversa donde debe expresarse adecuadamente como un retardo. Introduciendo `exp(-2*s)`, la Symbolic Toolbox no tiene ningún problema en manipularlo y dar el resultado correcto en `partfrac` o `ilaplace`.


Obviamente, todas las opciones son equivalentes y obtienen resultado idéntico.

Las ideas de este vídeo, si se hacen simbólicamente sobre un proceso $\tau \frac{dy}{dt} + y = Ku$ dan lugar a las fórmulas teóricas del vídeo [136], populares en muchos libros de texto por su utilidad para describir rápidamente los transitorios de este tipo de procesos.

A veces, los retardos son consecuencia de formas de onda de entradas “a trozos”, como en el ejemplo resuelto en el vídeo [129] (entrada pulso rectangular).

Problemas genéricos de entradas definidas a trozos se abordan en el vídeo [125].

Circuito de 1er orden ante tren de pulsos senoidales

[122: sinpulRCR]	Respuesta temporal circuito RCR ante pulso senoidal y condiciones iniciales no nulas (1: Laplace, superposición) Materiales: [CÓD.: PulsoSenoidalRCR.mlx] [PDF]	**  13:51 [YouTube ►]
------------------	---	---

* [LINK TO ENGLISH VERSION](#)

Este vídeo calcula la respuesta temporal, ante un pulso senoidal, de un circuito con 1 condensador y dos resistencias, por el método de la transformada de Laplace. UN ejemplo similar ante pulso “rectangular” puede verse en el vídeo [129].

Los tres primeros minutos discuten rápidamente el modelado en representación interna (dominio temporal) y en función de transferencia del sistema (también se discute el término de condiciones iniciales). Una versión ligeramente más extendida de la fase del modelado se puede encontrar en el vídeo [??] (en inglés).

El pulso senoidal y su transformada se discuten en el vídeo [128], cuya visualización se aconseja. No obstante, esta transformada NO va a ser usada aquí, porque la misma idea de retraso y superposición que se usó en ese vídeo aquí se usará para retrasar y superponer la salida ante una senoide completa.

La parte final del vídeo discute la respuesta ante condiciones iniciales no nulas y el mismo pulso de entrada. Obviamente, se trata de sumarle simplemente una exponencial (transformada inversa de término de condiciones iniciales, respuesta libre) a la solución anteriormente calculada.

Una forma alternativa de cálculo de la respuesta “a trozos” para este mismo circuito se discute en el vídeo [123], donde no se utiliza el concepto de “retardo” aquí explotado; además, la respuesta ante un tren de pulsos senoidales que se repite indefinidamente se discute en el vídeo [124].

Ejemplos adicionales de la metodología aquí empleada se abordan en los vídeos [125], [126] y [127].

[123: **sinpulRCR2**]
12:43

Respuesta temporal circuito RCR ante pulso senoidal y condiciones iniciales no nulas (2: por tramos)

Materiales: [[Cód.:](#) PulsoSenoidalRCR.mlx] [[PDF](#)]

[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo calcula la respuesta temporal de un circuito con 2 resistencias y 1 condensador ante un pulso senoidal (semiperíodo) y condiciones iniciales no nulas. El mismo problema fue abordado en el vídeo [122], donde se usaba la idea de “linealidad” e “invarianza temporal” para poder expresar la salida por superposición de dos respuestas ante senoide una de ellas retardada. El planteamiento del problema y el resumen del referido vídeo se esbozan en los primeros 2 minutos y medio de este material.

Aquí, se resuelve el mismo problema “por tramos”, prescindiendo de los conceptos de *retardo* y *superposición* del vídeo arriba referido. Se usan dos conceptos principales, discutidos “filosóficamente” en el vídeo [9]:

- La *causalidad*: si el pulso y la senoide completa coinciden durante los primeros 0.01 segundos, también coincidirán sus salidas en dicho intervalo (la salida no depende de cambios futuros en la forma de onda de entrada).
- El *concepto de estado*: todo el pasado se resume en el estado (carga del condensador) en el momento en que la entrada de ja de ser senoidal.

Estas dos ideas permiten resolver dos problemas de ecuaciones diferenciales, uno para el intervalo $[0, 0.01]$ y otro para $t \geq 0.01$, siendo la condición inicial del segundo intervalo igual a la condición “final” del primero.

La repetición de esta idea a muchos intervalos será utilizada para calcular la respuesta a un tren de pulsos senoidales en el vídeo [124].

[124: **trenpulRCR**]
15:45

Respuesta temporal de un circuito RCR ante tren de pulsos senoidales

Materiales: [[Cód.:](#) TrenPulsoSenoidalRCR.mlx] [[PDF](#)]

[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo calcula la respuesta ante un tren de pulsos senoidales de un circuito con 2 resistencias y 1 condensador. Forma parte del caso de estudio de circuito ante pulso senoidal constituido por los vídeos [128], [122], [123] y éste.

El desarrollo está basado en la superposición de trozos de respuesta calculados por tramos, por los conceptos de causalidad y de estado, de un modo análogo al usado en el vídeo [123], cuya visualización previa se recomienda para poder comprender mejor lo que se hace aquí.

Primero, se justifica que la transformada de Laplace de una señal periódica es la de la señal en un período dividida por $(1 - e^{-Ts})$, con fórmulas de suma de progresión geométrica.

No obstante, esa transformada de Laplace del tren de pulsos no es utilizada en el desarrollo, porque su inversa resulta complicada de abordar. Se usa resolución por tramos de 10 ms, usando como condición inicial del siguiente tramo la condición final del previo.


La parte final del vídeo discute el régimen permanente (no senoidal, porque la entrada no lo es: existirán armónicos), calculándolo mediante resolución de la ecuación de la respuesta que iguala $y(t)$ con $y(t + 0.01)$, siendo la incógnita de la ecuación la condición inicial.

NOTA: esta NO es la respuesta del circuito ante un rectificador que tenga, por ejemplo, un diodo que evite la circulación de intensidad en sentido inverso. Esto es un mero ejemplo académico de transformada de Laplace y no una presentación rigurosa de las señales rectificadas en electrónica de potencia.

Más ejemplos de formas de onda por tramos

[125: pwlap1] Respuesta temporal con entrada a tramos: planteamiento general, ejemplo tren escalones (superposición)

Materiales: [RTformasondatramos.pdf]

**  12:53

[YouTube ►]

Este vídeo plantea el problema de calcular la salida de un sistema ante una forma de onda a tramos con forma de ‘rampa’ o ‘escalón’.


Complementa este vídeo con el detalle con Laplace de un ejemplo completo que aparece en el vídeo [121].

Calculado por ‘superposición’ significa que la respuesta a dicha forma de onda se calculará a partir de las respuestas ante escalón y rampa adecuadamente escaladas y retrasadas según lo estén en la entrada. El ‘truco’ es descomponer la entrada en una suma de funciones que estén en tablas de transformadas (pero para ello dichos sumandos deben ser cero antes de un cierto instante y luego tener ‘la fórmula de las tablas’ hasta tiempo infinito).

Un primer ejemplo aplica la idea a una forma de onda de dos escalones. Los vídeos [126] y [127] ahondan en este caso de estudio y consideran formas de onda con rampas o, en caso discontinuo, con rampas y escalones mezclados (el segundo de ellos).

[126: pwlap2] Respuesta temporal: entrada a tramos, ejemplo tren de rampas

Materiales: [RTformasondatramos.pdf]

**  09:09

[YouTube ►]


Este vídeo continua con el cálculo de respuesta ante formas de entrada a tramos de un sistema lineal invariante en el tiempo, con ejemplos adicionales al tren de escalones considerado en el vídeo [125] previo a éste.

Aquí se considera una forma de onda de la señal $u(t)$ que puede descomponerse como suma de tres rampas, con lo que la respuesta temporal $y(t)$ será la suma de tres respuestas ante rampa adecuadamente escaladas y retrasadas.

El vídeo [127] continua el caso de estudio con una forma de onda que requiere la superposición de escalones y rampas porque hay cambios de pendiente y discontinuidades.

[127: pwlap3] Respuesta temporal entrada a tramos, ejemplo rampas y escalones combinados

Materiales: [RTformasondatramos.pdf]

***  08:35

[YouTube ►]

Este vídeo continua el caso de estudio de respuesta temporal ante entradas definidas a tramos de escalón y rampa con un último ejemplo que complementa a los vídeos [125] y [126] que deberías visualizar previamente.

Aquí manejamos una forma de onda que requiere la superposición de escalones y rampas porque hay cambios de pendiente y discontinuidades.

Respuesta temporal ante entradas definidas por tramos (piecewise)

La transformada de Laplace de entradas definidas “por tramos” incorpora retardos en cada “cambio de fórmula” de $u(t)$, como en el siguiente ejemplo:

[128: sinpulL] Transformada de Laplace de un pulso senoidal (semiperíodo)

Materiales: [[CÓD.:](#) PulsoSenoidalRCR.mlx] [[PDF](#)]

*** 11:33

[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo presenta metodologías de cálculo de la transformada de Laplace de un pulso senoidal (un semiperíodo de la función seno).

Aparte de calcularla con el comando `laplace` de la Symbolic Toolbox de Matlab, plantea calcularla (a) como la superposición de una senoide y de la misma senoide retrasada un semiperíodo usando el operador retardo e^{-ds} , o (b) directamente haciendo la integral que define la transformada de Laplace, por partes 2 veces.

Obviamente, todos los resultados son idénticos.

El vídeo [129] presenta resultados similares para un pulso constante como superposición de dos escalones.

NOTA: aunque este tipo de preguntas son típicas en “exámenes de grado”, resulta que aplicando “superposición”, muchas veces NO es necesaria la transformada de Laplace de la entrada calculada aquí para calcular respuesta temporal, dado que, por ejemplo, podríamos calcular la respuesta a una senoide (completa, de $t = 0$ a infinito, transformada $\omega/(s^2 + \omega^2)$) y aplicar la misma superposición en la salida. De hecho, así se hace en bastantes de los ejemplos con entradas “por tramos” en otros vídeos.

El pulso senoidal como entrada a un circuito eléctrico es estudiado en el vídeo [122].

En ocasiones, la transformada de Laplace de una entrada piecewise nunca es “explícitamente” utilizada, porque la salida se puede obtener por superposición. El siguiente video ejemplifica el caso de un pulso rectangular de duración finita.

[129: dly1erpul] Respuesta temporal primer orden inestable ante pulso rectangular (transformada de Laplace)

Materiales: [[CÓD.:](#) PartFracDemoESDelay.mlx] [[PDF](#)]

*** 19:38

[YouTube ▶]

Este vídeo obtiene, por métodos de transformada de Laplace, la respuesta ante un pulso rectangular del sistema de primer orden inestable $\frac{dy}{dt} - y(t)/3 = u(t)$; por completitud, también se incluyen condiciones iniciales no nulas.

Las ideas fundamentales son similares a las del ejemplo más sencillo del vídeo [121] que, posiblemente, te interese visualizar antes que el presente vídeo.

La idea básica aquí es darse cuenta de que un pulso rectangular puede descomponerse como la suma de un escalón “hacia arriba” en el instante de inicio del pulso ($t = 0$ en este ejemplo) y otro escalón “hacia abajo” en el instante donde el pulso finaliza.

Una vez comprendida esa idea, la solución se aborda mediante dos métodos (equivalentes):

Primero, obtener la respuesta ante escalón sin retardo; a continuación utilizando linealidad e invarianza temporal, obtener la respuesta ante escalón descendente retrasado y, por el principio de superposición obtener la respuesta pedida como la suma de tres componentes (escalón ascendente, escalón descendente, respuesta libre ante condiciones iniciales).

La segunda opción es incluir el retardo en las manipulaciones en el dominio de Laplace, utilizando el operador retardo e^{-ds} que esta vez aparece en la transformada de Laplace $U(s)$ del pulso rectangular, en vez de en la transformada de la ecuación diferencial como en el vídeo [121], pero las manipulaciones son muy parecidas.

Obviamente, todas las opciones son equivalentes y obtienen resultado idéntico. Recuerda que las funciones simbólicas “a trozos” necesitan el comando Matlab `heaviside` de la Symbolic Toolbox para su manejo y representación gráfica adecuada.

Otro ejemplo de las mismas ideas (respuesta ante un tren de escalones, que no tiene forma de pulso pero que, básicamente, es muy parecido) se presenta en el vídeo [175] aplicado a un motor de corriente continua. También se presenta en el vídeo [128] el caso de un pulso senoidal.

Más ejemplos relacionados (entradas a tramos) se discuten en los vídeos [125], [126] y [127].

La idea también puede aplicarse a otras formas de onda, como rampas:

[130: dlyramp]

Respuesta temporal sistema segundo orden ante rampa truncada, por transf. Laplace; comparación con escalón

Materiales: [[CÓD.:](#) PartFracDemoESDelayRamp.mlx] [[PDF](#)]

**** 15:58

[YouTube ▶]

En este material se discute cómo calcular la respuesta de un sistema de segundo orden oscilatorio (subamortiguado, en la terminología formal) lineal ante una rampa que, tras un tiempo, se queda truncada a un valor constante. El modelo es en representación interna, pero se calcula la función de transferencia para usar transformada de Laplace en la obtención de la respuesta temporal.

La idea básica es expresar la rampa como suma de una rampa ascendente en $t = 0$ y otra descendente en $t = 10$, de modo que la suma se mantiene constante.

Aplicando ideas muy similares al caso de “pulso rectangular” discutido en el vídeo [129], aquí con $1/s^2$ en vez de $1/s$, se obtiene la respuesta, primero simplemente usando la idea de invarianza temporal y superposición y luego, alternativamente (pero más laborioso) utilizando el operador retardo e^{-ds} en las manipulaciones en el dominio de Laplace.

Tecnológicamente, se trata de una transición suave gradual entre dos puntos de funcionamiento, de modo que las oscilaciones durante el transitorio tienen menos amplitud que ante una entrada escalón (transición más brusca). Es habitual su uso, dado que los cambios “a gran velocidad” de variables (el escalón cambia a velocidad infinita) pueden excitar no-linealidades, y también incluso en procesos lineales, aumentar la sobreoscilación.

4.2.5 Matriz de transferencia (MIMO)

[131: mdt]

Representación en Matriz de Transferencia de sistemas multivariables

Materiales: [MdTPrt1.pdf]

** 10:27

[YouTube ▶]

Este vídeo describe cómo, a partir del concepto de superposición de sistemas lineales, se origina el concepto de matriz de transferencia $G(s)$ de dimensiones n . de salida \times n . de entradas en sistemas multivariables, generalizando el concepto monovariable de “función” de transferencia $G(s)$.

Básicamente, se trata de reescribir en forma de matriz el resultado de despejar las salidas a partir de la transformada de Laplace de las ecuaciones del modelo. No hay nada teóricamente diferente a las ideas de los ejemplos en los vídeos [169] o [181], cuya visualización se recomienda. Aquí se esbozan ejemplos sencillos de un sistema con 1 salida y 2 entradas (la matriz de transferencia es una fila), de otro sistema con varias salidas y una entrada (la matriz de transferencia es una columna) y de un sistema con tres salidas y 2 entradas (El detalle sobre el modelado físico, matriz de transferencia y representación interna de ese último ejemplo puede consultarse en los vídeos [167] y [168], respectivamente).

En control multivariable existe una “lucha” entre dominio de Laplace (MdT) y representación en variables de estado. La representación en matriz de transferencia tiene como ventaja que se basa en las mismas ideas de las funciones de transferencia, y que sus elementos son efectos “individuales” de cada entrada sobre cada salida. Por contra, la manipulación de matrices cuyos elementos son cocientes de polinomios es mucho más engorrosa que la manipulación de las matrices A, B, C, D de representación interna de sistemas lineales que son puramente numéricas. Se debe ser capaz de alternar entre ambas representaciones según convenga. Una discusión más detallada de estas ventajas e inconvenientes se aborda en el vídeo [268].

4.3 Teorema del Valor Final

[132: tvf1]

Teorema valor final (Laplace) [1]: enunciado, uso y ejemplo sencillo

*** 08:40

Materiales:

[\[Cód.: TMAValFin.mlx \]](#)
[\[PDF \]](#)

[\[YouTube ► \]](#)

Este vídeo enuncia el teorema de valor final en el dominio de Laplace, esto es, que $\lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sY(s)$. Plantea la intuición de que $s \rightarrow 0$ es parecido al concepto de equilibrio (derivadas a cero), y presenta un breve ejemplo de su aplicación motivando su utilidad.

En el ejemplo, se compara la obtención “rápida” del valor final de cierta salida comparada con la obtención del mismo de forma “laboriosa” con fracciones simples y tablas de transformadas (laboriosa en un examen de lápiz y papel, porque realmente la Symbolic toolbox de Matlab lo hace todo en dos líneas).

La demostración del teorema se discute en el vídeo [133], su relación con el concepto de “equilibrio” y “ganancia estática” en el vídeo [134] y más ejemplos de aplicación en el vídeo [135].

El teorema del valor final se usa profusamente para calcular el régimen estacionario en bucle cerrado de sistemas de control (error de posición, velocidad, etc.I, discutido en los vídeos [356] y [357].

[133: tvf2]

Teorema valor final (Laplace) [2]: enunciado y demostración

*** 07:28

Materiales:

[\[Cód.: TMAValFin.mlx \]](#)
[\[PDF \]](#)

[\[YouTube ► \]](#)

Este vídeo enuncia y demuestra el teorema de valor final en el dominio de Laplace, de dos formas alternativas: (a) por fracciones simples (resíduos) y (b) a partir de la propia definición de transformada de Laplace y sus propiedades.

Motivación y un primer ejemplo de uso se discuten en el vídeo [132], la relación con los conceptos de “equilibrio” y “ganancia estática” en el vídeo [134], y ejemplos de aplicación en detalle se abordan en el vídeo [135].

[134: tvf3]

Teorema valor final (Laplace) [3]: relación con ganancia y ecuaciones en equilibrio

*** 14:45

Materiales:

[\[Cód.: TMAValFin.mlx \]](#)
[\[PDF \]](#)

[\[YouTube ► \]](#)

Este vídeo discute la relación entre el teorema del valor final (enunciado y motivación en vídeo [132]; demostración en vídeo [133]) y el concepto de ecuaciones de un sistema realizable en equilibrio, tanto en representación interna como en forma de ecuación diferencial ordinaria. También introduce el concepto de “ganancia estática” de un sistema como corolario del teorema.

La diferencia entre “teorema de valor final” y “ecuaciones en equilibrio” es que el primero requiere estabilidad de un sistema, mientras que un equilibrio puede ser inestable. Por contra, el teorema de valor final puede ser usado ante entradas acotadas tipo rampa o polinomio, por ejemplo, mientras que el concepto de equilibrio (al menos en su acepción más intuitiva) implica que todos los estados internos del sistema tengan un valor constante.

Ejemplos de aplicación del teorema de valor final y de las ideas discutidas en este vídeo se discuten en el vídeo [135].

[135: tvf4] [Teorema valor final \(Laplace\) \[4\]: ejemplos Matlab + generalización a entradas polinomiales](#)

Materiales: [[CÓD.:](#) TMAValFin.mlx] [[PDF](#)]

*** 15:30

[YouTube ▶]

Este vídeo concluye el caso de estudio sobre el teorema del valor final y sus aplicaciones, iniciado en el vídeo [132] (motivación, enunciado), el vídeo [133] (demostración), y el vídeo [134] (relación con los conceptos de equilibrio y ganancia estática).

Aquí se presentan tres ejemplos Matlab:

- Valor final de un sistema estable ante entrada que a su vez tiene valor final (ya esbozado en vídeos anteriores), concepto de ganancia estática
- Sistema inestable (donde el teorema de valor final realmente no aplica, pero sí tiene un punto de equilibrio inestable)
- Sistema con un cero en el origen (derivador puro), ante entrada rampa; en este caso el teorema de valor final sí aplica, pese a que el concepto de equilibrio no esté tan claro porque las variables que intervienen no son constantes.

La parte final del vídeo generaliza la idea de la rampa y el derivador a un sistema genérico: si ese sistema tiene m derivadores (ceros en el origen) entonces tendrá un valor final cero ante entradas de tipo polinomio de grado MENOR a m , un valor final constante (finito) ante entradas de tipo t^m , y un valor final infinito ante entradas con términos de grado MAYOR a m .

La aplicación a sistemas de control del teorema se discute en los vídeos [356] [357] y [358].

4.4 Recapitulación: respuesta de sistemas sencillos ante entradas típicas

4.4.1 Sistemas de primer orden (+retardo)

[136: ord1teo] [Respuesta sistemas primer orden ante escalón: ganancia, constante de tiempo \(teoría\).](#)

Materiales: [[CÓD.:](#) PruebasOrden1Esc.zip] [[PDF](#)]

** 14:49

[YouTube ▶]

Este vídeo detalla el cálculo de la respuesta temporal ante escalón (cond. iniciales nulas) del sistema $\tau \frac{dy}{dt} + y = K \cdot u$, cuya función de transferencia es $G(s) = \frac{K}{\tau s + 1}$.

Ante escalón de amplitud A , resulta $y(t) = KA \cdot (1 - e^{-t/\tau})$. Se comprueba que en $t = \tau$, la salida vale el 63% del valor final, y que en $t = 3\tau$ vale el 95%, y en $t = 4\tau$ vale el 98% del valor final KA .

Un ejemplo gráfico sobre $7/(5s + 1)$ ilustra la respuesta.

Se comenta que para $G(s) = b/(s + a)$ tendríamos $K = b/a$, $\tau = 1/a$.

Una animación para distintos valores de k y τ para comprender mejor su significado aparece en el vídeo [137], cuya visualización se aconseja.

La parte final del vídeo discute el efecto de un posible retardo d (todo ocurre d segundos más tarde), y también se comprueba gráficamente todo sobre la simulación de $7/(5 \cdot s + 1) \cdot e^{-2.5s}$.

Por su importancia en los temarios de asignaturas introductorias, una explicación alternativa de estos conceptos se detalla en el vídeo [109], de otro canal YouTube... y seguro que hay más si se busca...

El vídeo [154], continuación/complemento de éste, enfatiza los conceptos asociados a este último caso de primer orden con retardo, y pone un primer ejemplo de identificación experimental de ese tipo de sistemas.

La respuesta teórica ante "rampa" de sistemas de primer orden se discute en el vídeo [138], continuación de éste.

Una interpretación/aplicación 'intuitiva' del significado de la respuesta de sistemas de primer orden se aborda en el vídeo [152].

[137: Ktau]

Sistema 1er orden: efecto parámetros K, tau en respuesta
escalón (animación ganancia y constante de tiempo)

**
06:35

[YouTube ▶]

Este vídeo completa el vídeo [136] con un amplio conjunto de simulaciones de sistemas con función de transferencia $G(s) = K/(\tau \cdot s + 1)$ variando los parámetros de ganancia K y constante de tiempo τ en ciertos intervalos. Para ser más autocontenido, los dos primeros minutos revisan la teoría (respuesta escalón de primer orden) y luego se realizan las simulaciones comparativas con una veintena de ganancias y constantes de tiempo diferentes. No hay materiales porque el código es muy ad-hoc para el ejemplo, poco generalizable.

[138: ord1ramp]

Respuesta sistemas primer orden ante rampa en función de
ganancia y constante de tiempo: teoría + ejemplo Matlab

Materiales: [CÓD.: PruebasOrden1Rampa.mlx] [PDF]

**
14:31

[YouTube ▶]

Este vídeo es la versión ante entrada rampa de la teoría del vídeo [136]. En efecto, la entrada es $u(t) = At$ a un sistema de función de transferencia $G(s) = K/(\tau s + 1)$, o sea $U(s) = A/s^2$ y la respuesta se calcula mediante la descomposición en fracciones simples y transformada inversa de Laplace (ayudados por la Symbolic Toolbox).

Realmente, el régimen estacionario ante rampa de un sistema arbitrario de función de transferencia $G(s)$ tiene de fórmula $G(0)t + \frac{dG}{ds}(0)$, como se demuestra en el vídeo [150] que, opcionalmente, podríais querer visualizar.


La segunda mitad del vídeo, importante, se dedica a describir "intuitivamente" el significado de la solución: la salida es como la proporcionalidad estática K , pero retrasada τ segundos, excepto los primeros 3τ segundos donde hay un transitorio exponencial, claro.

Por supuesto, si el sistema tiene retardo d , todo ocurre d segundos después. Un par de ejemplos ilustran las características de la respuesta. Se esboza la idea de que con todo esto se podría hacer identificación experimental si se consigue ajustar un K y un τ a partir de las pendientes y los retrasos observados en los datos de laboratorio, aunque con ruido, determinar cuándo acaba el retardo puro y comienza el transitorio exponencial resulta más difícil ante rampa que ante escalón.

4.4.2 Sistemas de segundo orden

[139: ord2moti]

Respuesta escalón sistemas de segundo orden: motivación, caso
 polos reales, coeficiente de amortiguamiento
Materiales: [[CÓD.:](#) RespOrden2Escalon.mlx] [[PDF](#)]

***  13:48
 [YouTube ▶]

Este vídeo plantea el problema de caracterizar la respuesta (escalón) de un sistema de segundo orden (sin ceros).

Aquí las raíces del denominador (polos) pueden ser reales o complejas.


En el primer caso (reales), como son exponenciales e^{-pt} o similar, no es oscilatoria; por tanto, se propone describir la respuesta mediante la ganancia estática y el tiempo de establecimiento con la fórmula de primer orden. El caso de polos reales se denomina *sobreamortiguado* en la “jerga” de automática.

Con raíces complejas (caso usualmente denominado *subamortiguado*), la respuesta ya contiene exponenciales que multiplican a senos y cosenos. Ello requiere más discusión. En este vídeo se presenta la forma $G(s) = \frac{K\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$, por conveniencia (realmente es equivalente a $G(s) = \frac{a}{s^2 + bs + c}$). Se comprueba que cuando $0 \leq \xi < 1$ las raíces son complejas, estables, y que cuando $|\xi| \geq 1$ son reales. Cuando $\xi \leq 0$ entonces el sistema es inestable.

El caso real o inestable no tiene más misterio (si un sistema es inestable, no hay nada más que decir, no funciona; si tiene polos reales, aplicamos la fórmula de primer orden). El caso oscilatorio, expresamente, es objeto de un segundo vídeo [140] detallándolo.

[140: ord2step]

Respuesta escalón sistemas segundo orden subamortiguados:
 sobreoscilación, frecuencia propia, tiempo pico, ...
Materiales: [[CÓD.:](#) RespOrden2Escalon.mlx] [[PDF](#)]

***  19:01
 [YouTube ▶]

Este vídeo centra su desarrollo en la respuesta escalón de sistemas de segundo orden subamortiguados (oscilatorios) sin ceros. El caso sobreamortiguado (polos reales) ya se discutió en el vídeo [139]: básicamente, con ganancia y tiempo de establecimiento igualando la exponencial más lenta a, pongamos, 0.02, el problema *no oscilatorio* estaba resuelto casi como si fuera un sistema de primer orden.

Aquí se calculan diferentes parámetros de la respuesta ante escalón de amplitud A de $G(s) = \frac{K\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$, cuyo cálculo de polos también había sido detallado en el citado vídeo anterior a éste.


En concreto, con $0 < \xi < 1$, se comprueba que el valor final es $K \cdot A$, que el tiempo de establecimiento es, aproximadamente, $\pi/(\xi\omega_n)$, que la frecuencia de oscilaciones es $\omega_n\sqrt{1-\xi^2}$, y que la sobreoscilación (valor máximo menos final, todo dividido por valor final) es $e^{-\xi\pi/\sqrt{1-\xi^2}}$.

Un ejemplo numérico ilustra dichos resultados, analizando su respuesta. El efecto de los diferentes parámetros se estudia en detalle en el vídeo [141].

El paso inverso, aplicar las fórmulas para ajustar un modelo dado un registro experimental, se denomina identificación experimental y se discute en otros materiales, en concreto los vídeos [157] y [158].

[141: paramsor2]

Sistema segundo orden subamortiguado: efecto K, ω_n , ω_p ,
 sigma, amortiguamiento (simulación/animación)

**  19:56
 [YouTube ▶]

Este vídeo revisa brevemente la teoría y realiza diferentes simulaciones de un sistema oscilatorio estable de segundo orden para comprender el significado de diferentes parámetros. Otra revisión teórica algo más detallada puede consultarse en los vídeos [139] y [140].

Primero, simulamos la parametrización: $G(s) = \frac{K\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$ en tres casos: variar K , variar ξ o variar ω_n (dejando el resto de parámetros constantes).


Luego, simulamos la parametrización: $G(s) = \frac{K(\sigma^2 + \omega_p^2)}{(s + \sigma)^2 + \omega_p^2}$ donde $\omega_p = \omega_n \sqrt{1 - \xi^2}$ o, bueno, $\omega_n = \sigma^2 + \omega_p^2$. Se hacen dos ejemplos: un primero variando σ y manteniendo constante ω_p , y un segundo variando ω_p y manteniendo σ .

La primera parametrización tiene relación con módulo y argumento de las raíces (ω_n es el módulo, ξ es el coseno de 180-argumento); en la segunda σ es la parte real de las raíces (valor absoluto) y ω_p es la parte imaginaria. Ambas parametrizaciones son equivalentes, y es obvio cómo pasar de una a otra.

4.4.3 Efecto de los ceros en la dinámica de los sistemas

[142: zeros1]

Efecto de ceros en respuesta de un sistema: inversión de
dinámica, superposición con derivada, ceros en el origen
Materiales: [[CÓD.:](#) EfectoCEROS.mlx] [[PDF](#)]

***  15:25
[YouTube ▶]

Este vídeo discute el significado de los *ceros* en la respuesta de un sistema dinámico.

Como se trata de contenido para asignaturas de grado, se discute sólo el significado mono-variable y en representación de función de transferencia: raíces del numerador. El caso multivariable (ceros de transmisión, ...) se esboza en el vídeo [272].

Primero, se hace una discusión genérica sobre por qué podrían interesar.

La primera razón es el uso de modelos en *control*... Transformar la dinámica $G(s)$ de un sistema en otra dinámica (de “bucle cerrado”) usualmente más rápida, que llamaremos modelo deseado $M(s)$, si se considera de una forma “abstracta”, requiere “invertir” el sistema de modo que en alguna de las ecuaciones del control basado en modelos aparecerá algo como $Q(s) = G^{-1}(s) \cdot M(s)$, para que $G(s) \cdot Q(s) = M(s)$. Por tanto, los ceros de G serán polos de G^{-1} que aparecerán en $Q(s)$ y podrían, por tanto, ser de interés. Realmente el control no se hace así, pero es un marco teórico abstracto... sólo en el control por modelo interno (ver vídeos de la colección, por ej., el vídeo [391]) se hace *explícitamente* así.


La segunda idea es que un cero $G(s) = (\rho s + 1)G_1(s)$ indica una superposición de la salida de G_1 y su derivada sG_1 . Ello produce deformaciones en la respuesta.

En efecto, la última parte del vídeo analiza el significado de los ceros en el origen, numeradores $N(s) = s \cdot N_1(s)$. Como multiplicar por s en Laplace es derivar en el dominio del tiempo, se asocian a un derivador puro, obviamente. Dos ejemplos físicos de intensidad en circuito RC serie y de posición y velocidad de un sistema mecánico ilustran la idea.

El concepto de grado relativo (diferencia entre número de polos y de ceros) se discute en el vídeo [143], continuación de éste.

[143: zerosgr]

Ceros de una función de transferencia: grado relativo, efecto en
respuesta temporal y en frecuencia
Materiales: [[CÓD.:](#) EfectoCEROS.mlx] [[PDF](#)]

****  15:32
[YouTube ▶]

Este vídeo continúa con el análisis del efecto de los ceros (raíces del numerador de una FdT) en la respuesta de un sistema, que comenzó en el vídeo [142].

En este vídeo se define el concepto de *grado relativo*, igual a número de polos menos número de ceros de $G(s)$; obviamente, tiene que ser cero o positivo para que $G(s)$ sea realizable.

Se justifica intuitivamente que los sistemas de grado relativo cero tienen un salto brusco ante entrada escalón y que, en general, un sistema de grado relativo m tiene un salto brusco en la m -ésima derivada cuando se le introduce una entrada escalón... de hecho esa definición del salto en cierta derivada es la que es la "buena" y que puede extenderse a sistemas no lineales.


Esa idea puede usarse en control, pensando que nuestra entrada puede modificar la m -ésima derivada de cierta salida controlada, si la trayectoria no fuera correcta. La idea se denomina en control no lineal como *linealización por realimentación*, y se desarrolla, por ejemplo, en los vídeos [505] y [502].

La parte final del vídeo discute la relación del grado relativo con la pendiente del diagrama de Bode a altas frecuencias, que debe ser horizontal o descendente para que el sistema sea realizable.

► *Efectos de la adición de un cero:*

[14:33]
[Enlace]

[144: zerosord1] Efecto de ceros en respuesta temporal: sistemas de primer orden con cero adicional (fase mínima y no mínima)

**  12:47


Materiales: [Cód.: EfectoCEROS.mlx] [PDF]

[YouTube ▶]

Este vídeo analiza el efecto de los ceros en un sistema de primer orden, de tipo $\frac{K(\alpha \cdot s + 1)}{\tau \cdot s + 1}$. En concreto, se estudia la respuesta ante escalón. Los conceptos teóricos sobre ceros se discuten en los vídeos [142] y [143], de visualización aconsejable.

Como es de grado relativo cero, se tiene un salto brusco en la respuesta, en el mismo sentido que K si $\alpha > 0$, y en sentido contrario (fase no mínima) si $\alpha < 0$. El "cero" está en $z = 1/\alpha$, esto es, el sistema puede considerarse $\frac{K(\frac{1}{z} s + 1)}{\tau \cdot s + 1}$.

[145: expPZ] Respuesta experimental circuito polo-cero (RC+C)


**  03:40

Antonio Barrientos (UPM)

[PLAY ▶]

Este vídeo presenta un circuito con un RC en paralelo en serie con otro condensador, que tiene una función de transferencia de primer orden (un polo) con un cero. Lo construye experimentalmente y presenta su respuesta ante un tren de ondas cuadradas para comparar con la teoría.

[146: zerosord2] Efecto de ceros en respuesta temporal: sistema segundo orden subamortiguado con cero adicional (fase mínima y no mínima)

**  09:30

Materiales: [Cód.: EfectoCEROS.mlx] [PDF]

[YouTube ▶]

Este vídeo presenta el análisis (por simulación) de la respuesta temporal de un sistema con función de transferencia de segundo orden oscilatorio (subamortiguado) con cero adicional, tanto en casos de fase mínima como de fase no mínima. El resultado es que el tiempo de pico se adelanta en caso fase mínima, y se retrasa en fase no mínima. En ambos casos, la sobreoscilación aumenta. En el caso de fase no mínima, hay un efecto inicial de respuesta inversa, como de costumbre cuando hay un cero simple de ese tipo.


El efecto de los ceros sobre un sistema de primer orden se discute en el vídeo [144].

Sistemas de orden superior:

Los polos y ceros asociados a constantes de tiempo más rápidas pueden ser *grosso modo* despreciados para hacerse una idea cualitativa de la respuesta temporal.

[147: ordsupej1]

Análisis de respuesta escalón (aproximada) orden superior:
 ejemplos Matlab (1: polos reales)

**  13:48

Materiales: [[CÓD.:](#) PruebasOrdenSuperior.mlx] [[PDF](#)]

[YouTube ▶]

Este vídeo discute cómo analizar el comportamiento de sistemas de orden superior en base a ejemplos Matlab. La idea básica a aplicar aquí es la “dominancia”, donde una exponencial que dure “mucho” comparada con el resto será la más perceptible en la respuesta escalón una vez haya pasado el tiempo de establecimiento de las que duran menos; esa exponencial se denomina “dominante” y, al menos para hacerse una primera idea aproximada del comportamiento del sistema, podríamos aplicar las fórmulas de primer orden (vídeo (136)) a dicho polo dominante.


La idea se ilustra en un primer ejemplo de un sistema con varios polos reales, donde se concluye sólo mirando la función de transferencia que el transitorio será no oscilatorio, que el tiempo de establecimiento será similar al del polo dominante, y que haciendo $G(0)$ se obtiene la ganancia estática.

Un segundo ejemplo discute el caso totalmente opuesto a la dominancia: todos los polos (múltiples) en el mismo sitio, $G(s) = 1/(\tau s + 1)^n$. Se comprueba que cada polo extra añade un poquito más del 50

Más ejemplos que aparecen en el fichero mlx adjunto se explican en los vídeos [148] y [149].

[148: ordsupej2]

Análisis de respuesta escalón (aproximada) orden superior:
 ejemplos Matlab (2: mezcla reales/complejos)

***  13:50

Materiales: [[CÓD.:](#) PruebasOrdenSuperior.mlx] [[PDF](#)]

[YouTube ▶]

Este vídeo continua los ejemplos Matlab para ilustrar el concepto de dominancia que, con polos únicamente reales, se introdujo en los ejemplos del vídeo [147].

Aquí se presentan dos ejemplos de análisis de respuesta: un primer ejemplo con polo dominante real y polos dominados complejos conjugados (oscilatorios); un segundo ejemplo con polos dominantes un par subamortiguado, y polo dominado real.


En el segundo de los casos, se propone una aproximación eliminando los polos dominados para poder determinar (aproximadamente) la sobreoscilación y el tiempo de pico.

Como observación opcional, se argumenta que los polos eliminados tienen un comportamiento similar a un retardo. En otras referencias bibliográficas hay discusiones sobre sobreoscilación y tiempo de pico en sistemas de segundo orden con ceros y polos adicionales, pero al final, en vez de memorizar más fórmulas sobre amortiguamiento, etc. considero más recomendable al menos en una primera asignatura de grado simplemente simular a ver qué pasa.

Más ejemplos que aparecen en el fichero mlx adjunto se explican en el vídeo [149].

[149: ordsupej3]

Análisis de respuesta escalón (aproximada) orden superior:
 ejemplos Matlab (3: ceros, casos sin dominancia clara)

***  13:54

Materiales: [[CÓD.:](#) PruebasOrdenSuperior.mlx] [[PDF](#)]

[YouTube ▶]

Este vídeo es el último de los ejemplos Matlab del fichero mlx que os podéis descargar en el enlace. Los ejemplos 1 al 4 se detallan en los vídeos [147] y [148]. Aquí se explican los ejemplos 5 y 6.

En concreto, usando el concepto de dominancia, esboza la respuesta escalón de un sistema de orden 3 y con un cero, aproximándolo a un sistema de segundo orden que tiene únicamente el par de polos (complejos conjugados) dominantes.


Un último ejemplo donde hay un montón de polos y ceros cercanos, donde ninguno de ellos “domina” al resto, ni ninguno “cancela” a nada, pues entonces las fórmulas pensando en una aproximación de primer o segundo orden no valen... Simulamos y punto, a ver qué sale en la respuesta.

Este tipo de simplificaciones es la base de la reducción modal y de los conceptos de dominancia y cancelación discutidos en la Sección 9.1; también pueden aproximarse a un retraso (vídeos [fopd(10:40)] y [ordnafopd(10:25)]). Los detalles sobre todos esos enfoques a la reducción/simplificación de modelos de sistemas dinámicos no son objetivo de este capítulo.

Régimen estacionario ante rampa de una FdT

[150: regestramp1]
11:55

Respuesta estacionaria ante rampa de sistemas genéricos en función de transferencia (1): teoría

*** 

Materiales: [[Cód.:](#) regEstacRampa.mlx] [[PDF](#)]

[\[YouTube ▶\]](#)

Este vídeo presenta cómo calcular el régimen estacionario $P + Qt$ ante una entrada rampa (unitaria $u(t) = t$) en un sistema lineal con función de transferencia $G(s)$ arbitraria, generalizando el resultado del vídeo [138] que discutía el caso particular de sistema de primer orden (más sencillo, y recomendado ver antes que éste si estás iniciándote en estos temas).


Se demuestra, de dos formas (por descomposición en fracciones simples o por serie de Taylor), que si $G(s)$ no tiene polos en el origen, entonces $Q = G(0)$ y $P = G'(0)$ donde G' es la derivada de $G(s)$ respecto a s .

El caso más general, que unifica respuesta ante escalón, rampa o senoide en representación interna, se discute en el vídeo [165].

Interpretación y ejemplos de los resultados teóricos de este vídeo son discutidos en el vídeo [151], continuación de este.

[151: regestramp2]
13:18

Respuesta estacionaria ante rampa de sistemas genéricos en función de transferencia (2): interpretación y ejemplos

*** 

Materiales: [[Cód.:](#) regEstacRampa.mlx] [[PDF](#)]

[\[YouTube ▶\]](#)

Este vídeo es continuación del vídeo [150], donde se probó que el régimen estacionario ante rampa $u(t) = t$ de un sistema lineal de función de transferencia $G(s)$ es $G(0) \cdot t + G'(0)$ siendo G' la derivada con respecto a s de $G(s)$.

Aquí se discute su interpretación, que puede ser de dos formas: como “offset” desplazamiento vertical $G'(0)$; o como “retardo” desplazamiento temporal $G(0) \cdot (t + \delta)$ siendo $\delta = \frac{G'(0)}{G(0)}$.

A continuación, se ponen ejemplos Matlab de sistemas de primer orden y de segundo orden, tanto numéricos como simbólicos $G_1(s) = K/(\tau s + 1)$, $G_2(s) = K\omega_n^2/(s^2 + 2\xi\omega_n s + \omega_n^2)$. Por último se pone un ejemplo de sistema de tercer orden con ceros adicionales.

Caso de estudio linealidad + invarianza temporal + superposición

[152: linregla3]

Utilización intuitiva de dinámica lineal: tren de escalones con regla de tres, control bucle abierto

Materiales: [CÓD.: LinealIntuitivo.mlx] [PDF]

*
15:26

YouTube
▶

* [LINK TO ENGLISH VERSION](#)

Este video plantea utilizar un ensayo “escalón” obtenido experimentalmente (o en simulación, pero el detalle de qué ecuaciones y cómo es irrelevante mientras sean lineales –y aproximadamente de primer orden, sin “inercia”–) para hacer proporcionalidades (reglas de tres) orientadas a resolver los problemas siguientes:

1. Conseguir alcanzar un cierto punto de funcionamiento de referencia prefijado mediante UN escalón.
2. Conseguir alcanzar un cierto punto de funcionamiento de referencia prefijado en un tiempo de establecimiento dado, y luego mantenerse en él mediante DOS escalones sucesivos.

Para ello se hacen las proporcionalidades adecuadas y se comprueban en simulación las prestaciones alcanzadas con los perfiles de escalones calculados. Como el proceso simulado es un sistema lineal de primer orden, todo funciona perfectamente como se ha calculado. Si el proceso fuera de orden superior (con “inercia”) entonces podría haber sobrepasamientos y transitorios no esperados. Un ejemplo de ello con código prácticamente idéntico al aquí explicado se discute en el vídeo [153].

Esto podría ser considerado un caso especial de lo que se denomina *control en bucle abierto*: calcular un perfil de entrada sin usar sensores. En este caso sencillo, sólo linealidad y superposición son usados para calcular ese perfil de entrada, sin necesitar función de transferencia o representación en variables de estado como otras técnicas más avanzadas.

Ninguno de los cálculos ha necesitado ningún conocimiento teórico sobre linealización o ecuaciones diferenciales lineales, simplemente ha usado las proporcionalidades que caracterizan la respuesta de los sistemas lineales (incremental alrededor de un punto de operación).

[153: linregla3ord2]

Utilización intuitiva linealidad: tren de escalones regla de tres, proceso segundo orden (no va bien)

Materiales: [CÓD.: LinealIntuitivoOrden2.mlx] [PDF]

☰

YouTube
▶

08:16

* [LINK TO ENGLISH VERSION](#)

Este vídeo utiliza la misma metodología del vídeo [152] para calcular un tren de dos escalones que suba un proceso hasta un cierto valor deseado en un tiempo deseado, y que luego lo mantenga allí en equilibrio. Deberías visualizar el vídeo referido antes que este.

La única idea necesaria es linealidad/superposición, no se utiliza el modelo (por ejemplo, función de transferencia $G(s)$ o similar) en ningún momento para calcular el perfil de entrada.

El vídeo anterior que estamos comentando lo aplicaba a un proceso de primer orden, en el cual la técnica funcionaba perfectamente. Aquí se aplica a un proceso “secreto” que realmente es de orden 2, con lo que la “inercia” hace que cuando en el segundo escalón se conmute a la entrada que mantiene el punto de equilibrio deseado, haya un sobrepasamiento. El sobrepasamiento disminuye si se hace mas breve el primer escalón que aceleraba “extra” respecto al de equilibrio, pero entonces no se llega al sitio deseado con exactitud. Refinar esto, bien con un escalón adicional de “frenado”, bien en bucle cerrado (control PID), no entra, obviamente, en los objetivos de este vídeo que sólo pretende reforzar el concepto de “linealidad” de una dinámica.

4.4.4 Identificación experimental basada en respuesta ante entradas típicas

Sistemas de primer orden + retardo

[154: fopd] Modelos primer orden + retardo: propiedades resp. escalón e identificación experimental (manual) *** 10:40

Materiales: [Cód.: Identificar1ordenyretasoManual.zip] [PDF] [YouTube ►]

Este vídeo plantea las propiedades de la respuesta ante escalón de un modelo de primer orden + retardo $G(s) = \frac{k}{\tau s + 1} e^{-ds}$; una introducción al concepto de retardo puede verse en el vídeo [119]. Obviamente, son idénticas a las del sistema de primer orden sin retardo $G_0(s) = \frac{k}{\tau s + 1}$ pero retrasadas d unidades de tiempo (por ejemplo, los cruces típicos con el 63%, 95% y 98% del valor final que sin retardo ocurren en τ , 3τ y 4τ ahora ocurrirán en $d + \tau$, $d + 3\tau$ y $d + 4\tau$, respectivamente).

El detalle de la obtención de esa respuesta en sistemas de primer orden con/sin retardo se describe en el vídeo [136], cuya visualización previa podría ser aconsejable. También la respuesta en un ejemplo concreto numérico se aborda en los vídeos [120] y [121].

La segunda parte del vídeo carga un conjunto de datos experimentales y trata de ajustar “a mano” el modelo de primer orden + retardo superponiendo su simulación a los datos experimentales. La realización de esa tarea de forma automática puede hacerse con el comando **procest** de la System ID toolbox (ver vídeos [748] y [479]). Este tipo de modelos son populares en procesos industriales, y existen metodologías de diseño de PID adaptadas a ellos (por ejemplo, ver vídeo [403]). Si se deseara usar esas metodologías en procesos de orden superior, podrían aproximarse a modelos de esta clase basándose, por ejemplo, en el desarrollo de Taylor (vídeo [289]).

[155: ord1exg] Identificación experimental de sistemas de 1er orden (+retardo): tres ejemplos que salen perfectos ** 10:57

Materiales: [Cód.: PruebasOrden1Esc.zip] [PDF] [YouTube ►]

Este vídeo plantea ejemplos sencillos de aplicación de las ideas del vídeo [136] y [154].

Los tres ejemplos presentan un “supuesto” experimento ante escalón unitario (ajuste) y escalón descendente + chirp (senoide de frecuencia creciente). Al ajustar ganancia y constante de tiempo τ (cruce con 63% del valor final), se daría el modelo por identificado; luego se comprueba que los cruces con el 95% y 98% del valor final son en efecto en $t = 3\tau$ y $t = 4\tau$. Bueno, el tercer ejemplo incorpora retardo, pero también sale todo “clavado” como la teoría predice... porque el experimento realmente ha sido una simulación de un sistema de primer orden en los tres casos.

Otros ejemplos que no ajustan tan bien se discuten en el vídeo [154] y el vídeo [156].

[156: ord1exmal] Identificación experimental de sistemas de 1er orden (+retardo): ejemplos no tan perfectos *** 17:36

Materiales: [Cód.: PruebasOrden1Esc.zip] [PDF] [YouTube ►]

Este vídeo continua con ejemplos de aplicación de las ideas del vídeo [136] y [154].

El vídeo [155] planteaba tres ejemplos casi “perfectos” donde los cruces con 63%, 95% y 98% cuadraban con gran precisión con lo previsto por la teoría. Aquí se presentan un par de ejemplos donde hay que echar un poco más de imaginación, el ajuste no es tan perfecto, y hay que comprobar si el ajuste con datos de validación es o no aceptable. Claro, qué es “aceptable” y qué no lo es depende de la exactitud requerida en cada aplicación, cosa que aquí no consideramos. El objetivo es, simplemente, darse cuenta de que los datos experimentales pueden no ajustarse del todo bien a las fórmulas teóricas, pero no


hay ningún umbral “claro” que diga cuando el modelo sí ajusta a un primer orden más retardo o cuando no.

En el primero de los ejemplos (y material anexo) se usa `procest` de la System Identification Toolbox de Matlab para intentar ajustar los parámetros, con resultados parecidos al proceso “manual”. Para más detalles sobre un ejemplo adicional, ver vídeo [748].

Sistemas de segundo orden

[157: ord2idg]

Respuesta escalón sistemas de segundo orden subamortiguado:
identificación experimental (ejemplo 1)
Materiales: [[CÓD.:](#) IDexperimOrden2Escalon.zip] [[PDF](#)]

***  12:49
[YouTube ▶]


Este vídeo aplica las fórmulas de respuesta ante escalón de sistemas de segundo orden subamortiguados, discutidas en detalle en el vídeo [140], a la “inversa”, para hacer identificación experimental, esto es, dado un registro de una respuesta ante escalón, ajustarle una función de transferencia $K\omega_n^2/(s^2 + 2\xi\omega_n s + \omega_n^2)$ que replique el experimento.

Básicamente, a partir del valor final se obtiene K , a partir de valor máximo se tiene la medida experimental de la sobreoscilación $\delta = (max - fin)/fin$, con lo que una fórmula calcula ξ a partir de δ . La frecuencia de las oscilaciones ω_p se mide experimentalmente viendo el período (o el semiperíodo, que debería coincidir con el tiempo del primer pico); con ella se calcula $\omega_n = \omega_p/\sqrt{1 - \xi^2}$.

Este ejemplo está preparado para que la metodología funcione perfectamente, y en efecto, la respuesta escalón del modelo identificado se superpone casi a la perfección con el experimento. Otro ejemplo con datos más “realistas” con ruidos o dinámica adicional, donde ya el ajuste no será perfecto, se discute en el vídeo [158], continuación de éste.

[158: ord2idb]

Respuesta escalón segundo orden subamortiguado:
identificación experimental (2), comparación con procest
Materiales: [[CÓD.:](#) IDexperimOrden2Escalon.zip] [[PDF](#)]

***  16:32
[YouTube ▶]

Este vídeo aplica las fórmulas de respuesta ante escalón de sistemas de segundo orden subamortiguados, discutidas en el vídeo [140], a la “inversa”, para hacer identificación experimental, esto es, dado un registro de una respuesta ante escalón, ajustarle una función de transferencia $K\omega_n^2/(s^2 + 2\xi\omega_n s + \omega_n^2)$ que replique el experimento. El vídeo [??] también hacía lo mismo a partir de unos datos “preparados” para que todo ajustase perfectamente, se recomienda visualizarlo con anterioridad al que aquí discutimos.

Aquí detallaremos un segundo ejemplo con datos más “realistas” con ruidos y dinámica adicional. El ajuste no “perfecto” nos recomienda, posiblemente, una vez tengamos un modelo preliminar usando con más o menos suerte las fórmulas del vídeo [??], que abordemos un “ajuste fino” por prueba y error a ver si acertamos con algo que ajuste mejor al experimento.

Obviamente, el ajuste fino puede ser manual o bien dejarse en manos de un optimizador. Podríamos modificar fácilmente el código del vídeo [749] para minimizar un índice de ajuste de mínimos cuadrados o, vaya, dejarle al System Identification Toolbox que lo haga dado que ya implementa esas rutinas. En este caso, se utiliza el comando `procest` para ajustar un modelo ‘P2U’ (2 pole, underdamped) a ver si tiene más suerte que nosotros manualmente. También ajustamos un modelo ‘P2UZ’ incorporando ceros en el numerador. Al final, comparamos ambos modelos de procest y los modelos ajustados a mano, con los datos de laboratorio, y sacamos conclusiones.

Otros ejemplos de uso del comando `procest` en caso no oscilatorio pueden verse en el vídeo [748].

Identificación en bucle cerrado (introducción)

[159: idbcind1] Identificación en bucle cerrado: introducción, planteamiento del problema, identificación directa vs. indirecta *** 11:52

Materiales: [CÓD.: IdentBucleCerradoIndirectaOrd2.mlx] [PDF] [YouTube ►]

Este vídeo plantea el problema de la identificación en bucle cerrado, con datos que vienen de un bucle con cierto controlador conocido activo. Esto forzosamente ocurrirá, por ejemplo, en la mayoría de casos con procesos inestables... Si hay suerte, con un primer controlador se conseguirá estabilizar y, con la información obtenida, identificar un mejor modelo para diseñar un mejor control.

Se plantean dos posibilidades:

- identificación 'directa' a partir de $u(t)$ e $y(t)$ siendo $u(t)$ generada por el control ante cambios de referencia, por ejemplo... Un primer problema (salvable) es que aunque la referencia sea un escalón, $u(t)$ no lo será, por lo que necesitaremos software de optimización para ajustar el modelo. Un segundo problema es que con perturbaciones desconocidas, la entrada al proceso NO es la salida del controlador... en algunos casos, la identificación está 'sesgada' hacia la inversa del controlador.
- identificación 'indirecta', donde se identifica una función de bucle cerrado, por ejemplo $y(s) = T(s) \cdot r(s)$. Si $r(s)$ es un escalón, entonces podrían aplicarse ideas de constante de tiempo, sobreoscilación, o similares para una primera aproximación... Luego, a partir de $T(s) = GK/(1 + GK)$ podría despejarse $G(s)$. El problema podría ser que ajustar $T(s)$ por 'mínimos cuadrados' no daría el mismo resultado, que ajustar $G(s)$...

La identificación 'indirecta' es la más fácil para un primer curso introductorio y por eso el caso de estudio continuará explorando esa segunda opción.

Nota: estas ideas son muy básicas y poco precisas; la identificación en bucle cerrado es un problema de mayor dificultad que la identificación en 'bucle abierto' y las ideas esbozadas aquí son susceptibles de bastantes mejoras.

[160: idbcind2] Identificación indirecta en bucle cerrado ante ensayo escalón en referencia, ejemplo eje *** 14:49

Materiales: [CÓD.: IdentBucleCerradoIndirectaOrd2.mlx] [PDF] [YouTube ►]

Este vídeo plantea un problema de identificación indirecta de la Función de transferencia de un proceso (eje sujeto a pares) cuando existe un controlador conocido (identificación en bucle cerrado) a partir de un escalón en referencia y la medición de la salida (variable controlada).

En este ejemplo en concreto, usaremos las fórmulas de sistema de segundo orden subamortiguado para identificar $T(s) = GK/(1 + GK)$ y luego, a partir de $T(s)$ identificaremos como $G(s) = T/K/(1 - T)$.

[161: idbcind3] Identificación indirecta en bucle cerrado: ensayo escalón en acción control, ejemplo eje *** 13:02

Materiales: [CÓD.: IdentBucleCerradoIndirectaOrd2.mlx] [PDF] [YouTube ►]

Este vídeo plantea un problema de identificación indirecta en bucle cerrado ante un escalón en variable manipulada. El problema es muy parecido al del vídeo [160], que planteaba un escalón en la referencia e identificaba $T = GK/(1 + GK)$ para luego obtener $G(s) = T/K/(1 - T)$. Las ideas generales y motivación hacia este tipo de problemas se abordaron en el vídeo [159].

Aquí vamos a aplicar al proceso ‘lo que diga un regulador PD’ y añadiremos 1 Nm de excitación δu (una perturbación de par artificial, introducida por nosotros mismos intencionalmente y, por tanto, conocida para nuestros cálculos).

Con ello, el ensayo producirá $y(s) = W(s) \cdot r(s) = \frac{G}{1+GK} \cdot r(s)$. Usando las técnicas para identificación de sistema de 2o orden subamortiguado sin ceros, obtendremos un estimado de $W(s)$ y, con él, despejaremos $G(s) = \frac{W}{1-WK}$.

Nota: todo está muy ‘preparado’, y esto debe considerarse únicamente un “ejercicio académico”. Dependiendo de la estructura del regulador y del modelo del proceso, pueden salir otros tipos de funciones de transferencia de orden superior o con ceros, y las saturaciones, otras no linealidades, ruidos de proceso y de medida, etc. pueden hacer que este procedimiento de resultados bastante malos en algunas situaciones reales: la identificación en bucle cerrado no es un problema ‘trivial’ en la práctica.

4.5 Respuesta temporal en representación interna

4.5.1 Respuesta libre

[162: rtss1] Resolución $\dot{x}=Ax$: exponencial de matriz

Materiales: [RespTempss.pdf]


**  11:00
[YouTube ▶]

En este vídeo se plantea el problema de obtener la respuesta libre de un sistema como el de resolver $\dot{x} = Ax$ ante una condición inicial $x(0) = \xi$ arbitraria.

Se presenta el caso escalar $\dot{x} = -3x$ cuyas conclusiones permiten generalizar al caso $\dot{x} = Ax$ introduciendo la exponencial de una matriz (como una serie de potencias análoga a la exponencial escalar) y sus propiedades.

[163: rtssexpl] Fórmulas explícitas de la exponencial de una matriz (via forma de Jordan)

Materiales: [RespTempss.pdf]

***  10:32
[YouTube ▶]

Este vídeo discute cómo calcular de forma explícita la fórmula de la respuesta temporal (exponencial de matriz) de un sistema en representación interna $\frac{dx}{dt} = Ax$, discutida en el vídeo [162].

- Se revisa la forma canónica de Jordan de una matriz.
- Se calcula la exponencial de cada uno de los dos tipos de bloque de Jordan (diagonal, no-diagonal).
- Se analiza el caso de valores propios complejos, que da lugar a exponenciales de parte real y senos y cosenos de parte imaginaria en la respuesta temporal final.

[164: rtssml] Respuesta temporal (exponencial de matriz), ejemplo Matlab

Materiales: [Cód.: cosasresptemporal.mlx] [PDF]

**  10:38
[YouTube ▶]

En este vídeo se presenta código Matlab que ilustra los desarrollos teóricos de los vídeos [162] y [163] sobre la exponencial de matrices y su relación con la respuesta temporal de un sistema $\frac{dx}{dt} = Ax$.

Se utiliza el comando **expm** numéricamente para representar gráficamente la solución de la ecuación de estado y ecuación de salida de un sistema de orden 3.

Asimismo, se presenta el comando `expm` de la Symbolic Toolbox, que obtiene las expresiones explícitas de la exponencial de la matriz (elementos funciones del tiempo), discutidas en el vídeo [163].

La respuesta temporal ante escalón (forzada) de un sistema lineal también puede calcularse con exponenciales de matrices, ver vídeo [??].

4.5.2 Régimen Estacionario

[165: res] Régimen estacionario de sistemas en representación en variables de estado (repr. interna): constante/rampa/senoidal

Materiales: [RegEstacionarioDetermin.pdf]

*** 10:57

[YouTube ▶]

Este vídeo discute cómo calcular el régimen estacionario de sistemas en representación interna normalizada $\dot{x} = Ax + Bu$, ante entradas constantes, rampa o senoidales.

Las entradas constante/rampa/senoide se presentan como casos particulares de la salida de generadores lineales $\dot{\xi} = M\xi$, $u = N\xi$ marginalmente estables (valores propios de M en el eje imaginario).

El componente transitorio se calcula a partir de M y N de forma aislada, sin calcular la respuesta temporal completa, mediante la resolución de una *ecuación de Sylvester*.

Obviamente, el régimen estacionario ante una entrada constante es otro nombre para el concepto físico de “punto de equilibrio” $0 = Ax_{eq} + Bu_{eq}$, y el régimen estacionario ante una entrada senoidal es una reformulación del concepto de “respuesta en frecuencia” clásico.

[166: resm] Régimen estacionario constante/rampa/senoidal, repr. interna: ejemplo Matlab

Materiales: [Cód.: ejemploestacionario.mlx] [PDF]

*** 07:02

[YouTube ▶]

Este vídeo presenta código Matlab para realizar los cálculos sobre régimen estacionario en representación interna, cuya exposición teórica se ha presentado anteriormente en el vídeo [165].

En un sistema lineal de segundo orden con una entrada, se calculan los regímenes estacionarios constante/rampa/senoidal y se comparan con los obtenidos tras el cálculo de la respuesta temporal completa por el método de Laplace $\dot{x} = Ax + Bu$, $x(s) = (sI - A)^{-1} \cdot (Bu + x(0))$, viéndose que coinciden. Las transformadas de Laplace y sus inversas se evalúan usando la Symbolic Toolbox de Matlab.

4.6 Ejemplos de modelado y cálculo de respuesta temporal

[167: rcrl] Modelado de un circuito R-C-R con dos fuentes de tensión: matriz de transferencia, término de condiciones iniciales, resp. escalón

Materiales: [Cód.: ModeloRCR.mlx] [PDF]

*** 08:43

[YouTube ▶]

Este vídeo modela un circuito en T con dos resistencias y un condensador en cada una de las tres ramas, alimentado por dos fuentes de tensión.

Tras el modelado físico, se hace transformada de Laplace de las ecuaciones (incluyendo posibles condiciones iniciales no nulas), y se despejan la matriz de transferencia y el término de condiciones iniciales. Multiplicando la matriz de transferencia por $1/s$ se obtienen seis respuestas ante escalón individuales, y la antitransformada del término de condiciones iniciales obtiene la respuesta libre. Ejemplos numéricos de cálculo de respuesta temporal paso a paso con función de transferencia se pueden consultar en los vídeos [111], [112], [113].

El mismo problema es resuelto usando representación en variables de estado (representación interna normalizada) en el vídeo [168], cuya visualización se recomienda.

Un problema mecánico más complejo pero que conceptualmente sigue los mismos pasos se aborda en el vídeo [169].


[168: rcrssml]	Modelado/simulación de un circuito R-C-R con dos fuentes de tensión: representación interna	***  07:49
	Materiales: [CÓD.: ModeloRCR.mlx] [PDF]	[YouTube ►]

Este vídeo modela un circuito en T con dos resistencias y un condensador en cada una de las tres ramas, alimentado por dos fuentes de tensión.

Tras el modelado físico, se manipulan dichas ecuaciones para obtener la representación interna normalizada (ecuación de estado y ecuación de salida); al tratarse de un sistema lineal, se puede representar con $\frac{dx}{dt} = Ax + Bu$ (ec. estado), $y = Cx + Du$ (ec. salida), e introducirlo en la Control System Toolbox de Matlab mediante `ss(A,B,C,D)`.

El mismo circuito es modelado en el vídeo [167] mediante Transformada de Laplace y Matriz de Transferencia (MdT); los resultados de dicho vídeo se comparan con la fórmula que pasa de estado a MdT, $G(s) = C(SI - A)^{-1}B + D$, comprobando la coincidencia.


Estos modelos pueden ser usados para cálculo de respuesta temporal. Un ejemplo de circuito eléctrico ante pulso senoidal se discute en el vídeo [122].

[169: ep1]	Sistema engranajes y poleas (lineal). Modelado y Función de transferencia	**  19:37
	Materiales: [CÓD.: engranajepolea.zip] [PDF]	[YouTube ►]

Ejemplo de código para modelado de un sistema físico con Matlab:

- Este video describe el modelado de un sistema con un motor, engranajes, polea, masa, muelle y amortiguador. Un desarrollo conceptualmente parecido al aquí discutido pero sobre un modelo eléctrico mucho más simple (de primer orden) se aborda en el vídeo [167], cuya visualización previa a éste es muy recomendable.
- Utiliza el Symbolic Toolbox de Matlab para despejar las ecuaciones.
- Reemplazando derivadas por “s”, el mismo Toolbox representa el modelo en transformada de Laplace y despeja la función de transferencia de las salidas de interés. Cuando son varias salidas, se denomina *matriz de transferencia*, ver vídeo [131].
- Calcula la respuesta temporal, ante un par motor $\sin(3t)$, mediante el comando `ilaplace` del Symbolic Toolbox.

Otro ejemplo de motor eléctrico con reductora se aborda en el vídeo [173].

[170: ep2]	Sistema engranajes y poleas (lineal): representación en variables de estado	**/ ***  18:20
	Materiales: [CÓD.: engranajepolea.zip] [PDF]	[YouTube ►]

Este video continua el video [169], y obtiene la representación interna normalizada despejando con la Symbolic Toolbox de Matlab a partir de las ecuaciones (algebraico-diferenciales) de un modelo deducidas en el citado vídeo.

Inicialmente, se opera suponiendo que todas las variables que aparecen bajo el signo de la derivada son variables de estado... debería funcionar, pero no lo hace porque las ligaduras mecánicas hacen que los supuestos “estados” no sean *grados de libertad* mecánicos independientes. La metodología que en otros casos más sencillos sí funciona aquí no lo hace.

Desde un punto de vista abstracto, la razón de que no funcione es que las ecuaciones algebraico-diferenciales del modelo inicialmente escrito no son de índice 1, con lo cual se necesita añadir ecuaciones cinemáticas adicionales (derivadas de otras ecuaciones en el modelo inicial) para poder obtener la representación interna normalizada con éxito (y de un orden menor al inicialmente probado, dos veces el número de grados de libertad). ***

La segunda parte del vídeo analiza la respuesta del sistema:

- Utiliza el comando `step` para simular la respuesta ante escalón de dicha representación interna.
- Utiliza el comando `tf` para pasar de representación interna a función de transferencia. Comprueba la estabilidad examinando los polos.

[171: moll3free] Sistema 3 masas 4 muelles: análisis modal de la respuesta libre *** 14:50
Materiales: [[CÓD.:](#) ModosLibres3Masas4Muelles.zip] [[PDF](#)] [[YouTube](#) ►]

*[LINK TO ENGLISH VERSION](#)

Este vídeo presenta un análisis de los modos de la respuesta libre de un sistema de 3 masas y 4 muelles cuyo modelado se detalló en los vídeos [24], [25].

El código Matlab para visualizar una animación del movimiento de dichas masas se detalló en el vídeo [35]; este código será utilizado (con variaciones mínimas) también en este vídeo para presentar los resultados, pero su detalle no será explicado dado que ya se hizo en el vídeo referido.

Básicamente, los modos de la respuesta libre vienen dados por los autovalores y autovectores de la matriz A en una representación $\frac{dx}{dt} = Ax + Bu$, donde el término Bu no tiene ninguna relevancia en la respuesta libre aquí estudiada.

En este caso, son autovalores y autovectores complejos. La parte real de los autovalores (tasa de decaimiento) permite aproximar la duración del transitorio (tiempo hasta que las oscilaciones desaparecen); es la misma en todos los modos, aunque no tiene por qué serlo en un caso general. La parte imaginaria es la frecuencia de las oscilaciones en radianes por segundo.

Al final, la respuesta libre tiene tres modos oscilatorios que duran 65 segundos hasta desaparecer, de diferentes frecuencias y con las masas bien en fase bien en oposición de fase.

Se analizan autovalores y autovectores, y se carga como condición inicial la parte real de los autovectores para simular cada modo, y para visualizar una animación del mismo, para que se comprenda mejor el significado del modo. Estos modos son parecidos pero no iguales a los modos de “resonancia” (respuesta forzada ante entrada senoidal).

4.6.1 Caso de estudio: motor de corriente continua con carga mecánica

[172: motccmodel] Modelado motor corriente continua haciendo girar una carga (representación en variables de estado) ** 13:59
Materiales: [[CÓD.:](#) CasoMotorCC.mlx] [[PDF](#)] [[YouTube](#) ►]

Este vídeo modela un motor de corriente continua (ecuaciones básicas $E = k\omega$, $T = ki$) con inductancia (autoinducción L) y resistencia en los devanados, que hace girar una carga con una cierta inercia y un par de rozamiento proporcional a la velocidad angular.

El vídeo discute la generación de una representación en variables de estado de tercer orden, siendo el estado la posición y velocidad angulares, así como la intensidad que recorre la bobina.

Se observa que la velocidad angular y la intensidad, por sí solas, constituyen también una representación interna válida, de modo que si se pone la ecuación de la posición angular es porque se desea que sea una salida de mis análisis o simulaciones. Si no, podría reducirse a un sistema de orden 2 con estado (ω, i) , por ejemplo, en un entorno tecnológico donde la única variable de interés fuera la velocidad de giro del sistema electromecánico.

La segunda parte del vídeo considera la posibilidad de que la autoinducción L tienda a cero. Primero, se analiza el significado físico de eso (si la energía que se maneja en el campo magnético de la bobina es despreciable con respecto a la energía mecánica, ¿podría eliminarse la dinámica eléctrica?), y luego se modifica el modelo para hacer $L = 0$, obteniendo una nueva representación interna de orden 2 con estados (θ, ω) únicamente “mecánicos”. Obviamente, si no se quisiera saber nada de la posición angular, se reduciría a primer orden.

Se obtiene una representación matricial de ambos modelos lineales en representación interna, y se deja para vídeos posteriores la simulación comparativa para ver si, para ciertos valores de los parámetros, resulta lógico omitir la dinámica eléctrica o no. Aquí, simplemente hemos abordado el modelado de la dinámica del sistema en espacio de estados.

El caso de estudio continua en el vídeo [174] (función de transferencia, respuesta escalón con comando **step**), y en el vídeo [175] (respuesta ante escalón y ante tren de escalones por medio de transformada de Laplace).

El modelado de un motor eléctrico añadiendo un engranaje reductor se discute en el vídeo [173].

[173: moteng] Modelado motor CC con reductora

Materiales: [MotorEngranajeFinal.pdf]

***  18:54

[YouTube ▶]

Este vídeo modela un motor de corriente continua con un eje de inercia J_1 y un engranaje reductor 10 a 1 que conecta con un segundo eje de inercia J_2 .

Primero se revisa el modelado de un motor sin engranaje, simular al realizado en el vídeo [172], y su representación interna. Luego, dicho modelo se utiliza como parte de un modelo al que se le añadirán las ecuaciones de la dinámica del segundo eje.

Como la segunda velocidad angular es múltiplo de la primera, se reducirá la dinámica a una representación mínima sólo en función de intensidad de armadura y velocidad del eje 1. En las operaciones, aparecerá el concepto de momento de inercia equivalente (y coeficiente de rozamiento equivalente) de cómo se ve la carga en el eje 2 sobre el eje 1.

El vídeo finaliza con una representación normalizada en forma matricial de las ecuaciones de estado resultantes.

Otro ejemplo de sistema con engranajes y poleas se aborda en el vídeo [169].

[174: motcctfstep] Motor corriente continua haciendo girar carga: función de transferencia y respuesta escalón (step)

14:58

Materiales: [[CÓD.:](#) CasoMotorCC.mlx] [[PDF](#)]

*** 

[YouTube ▶]

Este vídeo obtiene la representación en el dominio de Laplace de los modelos en representación interna de un sistema electromecánico consistente en un motor de corriente continua que hace girar una carga. El detalle del modelado físico se discute en el vídeo [172], que se recomienda visualizar antes que éste. .

Aquí, se plantea cómo pasar de estado a función de transferencia (bueno, *matriz* de transferencia porque hay DOS salidas de interés, y por tanto debemos trabajar con dos funciones de transferencia simultáneamente, en forma de vector columna).

Obviamente, puede hacerse con manipulaciones de la Symbolic Toolbox (que deberían hacerse “manualmente” en un examen de lápiz y papel típico, práctico), pero también con el comando `tf` del Control System Toolbox. Se comprueba que todo coincide.

También se comprueba que al hacer $L = 0$ en la matriz de transferencia del modelo con dinámica eléctrica, el resultado coincide con la matriz de transferencia del modelo simplificado sin dinámica eléctrica de orden 2, como era de esperar intuitivamente.

La segunda mitad del vídeo obtiene la respuesta escalón con el comando `step` de la Control Systems Toolbox; un vídeo de próxima publicación lo hará paso a paso por el método de Laplace, pero no es relevante en este momento, donde sólo se busca “comprender” intuitivamente el sistema que estamos manejando.

Los objetivos de la parte final del vídeo son dos:

Primero, entender qué significa el transitorio (período de aceleración) y el régimen permanente (giro a velocidad constante de equilibrio, posición angular creciendo en forma “rampa”). Entender también que, en aplicaciones donde el primer segundo no sea importante, que el motor puede describirse de forma muy sencilla con una “proporcionalidad”: gira a una velocidad de 1.4 rad/s por cada Voltio de tensión de entrada; sólo cuando vamos a ver qué ocurre “dentro” del primer segundo es la dinámica importante.

El segundo objetivo es comprender las similitudes y diferencias entre el modelo de orden 3 con dinámica eléctrica y el modelo donde dicha dinámica no se tenía en cuenta... Se observa que, salvo a escalas de tiempo de unos 10 o 20 milisegundos donde sí hay gran diferencia, cuando las cosas se miran a escalas de tiempo de más de 200 milisegundos, no es de demasiada utilidad manejar un modelo complicado cuando un modelo más sencillo tiene una respuesta prácticamente idéntica; en teoría de sistemas, se dice que la dinámica eléctrica “no es dominante”.

Las fórmulas concretas de la respuesta ante escalón se obtendrán en otro vídeo, en concreto el vídeo [175], por medio de transformada de Laplace.

[175: motccLapl]

Motor corriente continua que gira carga: respuesta ante escalón
(método Laplace) y tren de escalones

Materiales: [[CÓD.:](#) CasoMotorCC.mlx] [[PDF](#)]

15:53

[\[YouTube ▶\]](#)

Este vídeo es continuación de los vídeos [172] (modelado físico) y [174] (función de transferencia, respuesta con Control Systems Toolbox).

En concreto, el objetivo de este vídeo es obtener mediante la transformada de Laplace, fracciones simples y transformada de Laplace inversa, la respuesta ante escalón de los modelos con y sin dinámica inductiva desarrollados en los vídeos arriba referidos, para parecerse a los desarrollos usuales en exámenes de “lápiz y papel” en primeros cursos de Automática

Se advierte de la necesidad, en algunos casos de usar la opción `FactorMode='real'` en el comando `partfrac` porque por defecto, busca polinomios con coeficientes racionales y, si los polos no son racionales, produce un resultado erróneo.

El resto de manipulaciones para la obtención de la respuesta ante escalón es bastante estándar, análogo a muchos otros vídeos.

La parte final del vídeo utiliza la respuesta ante escalón obtenida, llamémosle $\phi(t)$, para calcular la respuesta ante una entrada constante por tramos (tren de escalones), simplemente escalando y retrasando $\phi(t)$ del mismo modo que los sucesivos incrementos de valor de la señal de entrada. El razonamiento es similar a, por ejemplo, el del pulso rectangular en el vídeo [129] o la rampa truncada del vídeo [130].

4.6.2 Caso de estudio: calentamiento pieza de dos capas

En elaboración...

[176: term2mod1]

Modelado de calentamiento pieza de 2 capas: representación interna (espacio de estado)

Materiales:

**

13:35

[Cód.: termico2capascasoestudio.mlx]
[PDF]
[YouTube ►]

Este vídeo modela el calentamiento o enfriamiento de una pieza con dos capas (interior y exterior, temperaturas T_1 y T_2 respectivamente) introducida en un horno a temperatura T_{ext} .

Se hace un modelado en representación interna de orden 2 basándose en $MC_e \frac{dT}{dt} = Q$ y transmisión de calor proporcional a diferencia de temperaturas. Luego, se hace la representación matricial, considerando las temperaturas T_1 y T_2 como salidas.

Se analiza el equilibrio térmico, comprobando que se produce cuando todas las temperaturas se igualan y, por tanto, cualquier punto de operación considerando ‘cero’ una cierta temperatura base igual en todos los cuerpos, sería válido para operar con el modelo lineal. Las “derivadas a cero” en las ecuaciones, en forma matricial acaban indicando $y_{eq} = (-CA^{-1}B + D) \cdot u_{eq}$. Esta fórmula es importante y $(-CA^{-1}B + D)$ recibe el nombre de *matriz de ganancia estática*.

La parte final del vídeo plantea el interés de calcular y representar la diferencia de temperaturas $T_2 - T_1$, porque si es excesiva, las diferencias de dilatación podrían despegar la unión o fracturar la pieza por las tensiones mecánicas (bueno, si los materiales tuvieran un coeficiente de dilatación muy diferente, también se rompería la pieza al calentarla).

El modelado de orden 2 no es la única opción... dependiendo de la exactitud requerida, puede haber modelos más simples o más complicados, como se discute en el vídeo [177].

El caso de estudio continua en los vídeos: [178] (respuesta escalón) y [179] (respuesta rampa, tratamiento térmico suave para limitar diferencia de temperaturas).

[177: term2mod2]

Modelado de calentamiento pieza de 2 capas, opciones alternativas: 1er orden, EDP, elementos finitos

Materiales:

10:58

[Cód.: termico2capascasoestudio.mlx]
[PDF]
[YouTube ►]

Este vídeo, continuación del vídeo [176], plantea posibilidades alternativas del modelado de la pieza de 2 capas a calentar en un horno que se discutió en el referido vídeo.

Primero, se plantea que en ocasiones (si la capa externa es fina, tiene poca capacidad o/y la conductividad de la unión con T_1 o el exterior es alta) la temperatura de dicha capa T_2 sube rápidamente sin que la temperatura interna T_1 cambie mucho. Entonces, un modelo de primer orden podría ser razonable si el calentamiento “global” es “lento” (pongamos 15 minutos) y no nos interesa lo que le pase a T_2 en los primeros 10 segundos. De hecho, eso se observará en el vídeo [178](simulación ante escalón), aunque, por brevedad, no se detallará el tema de modelo más sencillo en el referido vídeo.

Suponiendo lo de arriba, se calculan las nuevas ecuaciones de estado y salida del modelo simplificado.

La segunda parte del vídeo (a partir del minuto [07:20]) esboza la idea que realmente podríamos pensar que cada una de las capas tiene “sub-capas” y podríamos modelar


un objeto de 5, 8, 10 o 300 capas... esos modelos se denominan modelos de *elementos finitos* y que realmente son una aproximación de las ecuaciones diferenciales en derivadas parciales (EDP) de transmisión de calor (en este caso, con simetría esférica, calculando una temperatura $T(t, r)$ función de tiempo y distancia al centro). Esta idea de las EDP es simplemente esbozada, dado que no es objetivo del caso de estudio, pero es importante tener presente que este tipo de opciones son muy usadas en materiales, transmisión de calor, etc.

El caso de estudio continua en los vídeos: [178] (respuesta escalón) y [179] (respuesta rampa, tratamiento térmico suave para limitar diferencia de temperaturas).

Si estás interesado en las EDPs en transmisión de calor, ejemplos de análisis en derivadas parciales de un calentador tubular aparecen en los vídeos [79] y [184].

[178: term2step] Calentamiento pieza de 2 capas: función (matriz) de transferencia, respuesta escalón control toolbox

Materiales: [[Cód.:](#) termico2capascasoestudio.mlx] [[PDF](#)]

***  15:44

[YouTube ►]

Este vídeo es continuación del caso de estudio de calentamiento de una pieza de 2 capas en un horno, cuyo modelado de segundo orden se discutió en el vídeo [176]. Opciones alternativas de modelado se abordan en el vídeo [177], aunque esos modelos alternativos no van a usarse aquí, por lo que no es necesario visualizar ese último vídeo.

Los primeros minutos del vídeo revisan el modelado y obtienen la función de transferencia (matriz, porque tenemos tres salidas $y_1 = T_1$, $y_2 = T_2$, $y_3 = T_2 - T_1$) mediante la fórmula matricial $C(sI - A)^{-1}B + D$ y mediante manipulaciones algebraicas “manuales”, con resultados idénticos.

A partir del minuto [04:40] se plantea el cómo ir desde FdT a una ecuación diferencial ordinaria... copiando coeficientes, puede escribirse una EDO cuya transformada de Laplace sea una FdT dada, pero se pierde la interpretación física que el modelo (A, B, C, D) sí que tenía.

A partir del minuto [07:30] se analizan los polos del sistema (exponenciales de la respuesta libre, raíces de denominador de funciones de transferencia).

Posteriormente, se introduce el modelo `ss(A,B,C,D)` en el Control Systems Toolbox, se comprueba que el comando `tf` devuelve la misma función de transferencia que las manipulaciones simbólicas de antes.


El comando `step` representa la respuesta ante escalón de las temperaturas y su diferencia (escalón significaría, por ejemplo, introducir la pieza rápidamente en un horno ya caliente).

Se observa que la temperatura 2 tiene unos primeros segundos de calentamiento muy rápido que hace que la diferencia de temperaturas suba mucho; luego ya hay otro modo “lento” de calentamiento hasta que todas las temperaturas se igualan a la del horno. Los autovalores y autovectores de la matriz A explican qué combinaciones de estados son las que son responsables de la dinámica “lenta” y “rápida” (esto se ve en el minuto [15:00] en las conclusiones finales).

Si la diferencia de temperaturas no es aceptable, debería hacerse un calentamiento “gradual” en “rampa”, como se discute en el vídeo [179] en detalle.

[179: term2ramp] Calentamiento pieza de 2 capas: tratamiento térmico en rampa para limitar diferencia de temperaturas

Materiales: [[Cód.:](#) termico2capascasoestudio.mlx] [[PDF](#)]

****  17:51

[YouTube ►]

Este vídeo concluye el caso de estudio de un tratamiento térmico a una pieza de 2 capas. El modelado se vio en el vídeo [176], como una representación interna de orden 2; opciones alternativas de modelado (no usadas aquí) se discuten en el vídeo [177], opcional.

El vídeo [178] obtiene la representación en función de transferencia en el dominio de Laplace (bueno, matriz de transferencia en rigor, porque hay tres salidas). También representa y interpreta físicamente la respuesta ante escalón (introducción brusca en horno caliente). La conclusión de ese vídeo es que la diferencia de temperaturas se considera excesiva y que será necesario un calentamiento gradual (rampa) hasta alcanzar 600 grados (objetivo tecnológico), y detallar todo eso es el objetivo principal del vídeo aquí descrito.

En concreto, tras revisar en los primeros dos minutos las conclusiones de los vídeos arriba referidos, representa la respuesta ante rampa mediante la Control Systems Toolbox; necesita un “truco”, que es usar el comando `step` pero del sistema multiplicado por un integrador $1/s$ que convierte el escalón que aplica `step` en una rampa.

Se comprueba que ante la rampa $T_{ext} = 1 \cdot t$ la diferencia de temperaturas llega a valer 110 grados, por lo que se requiere otra pendiente.

Ello motiva para obtener la fórmula de la respuesta ante rampa en transformada de Laplace, por si fuera necesaria para calcular el tratamiento térmico óptimo. El procedimiento es el usual ($FdT \cdot 1/s^2$, fracciones simples, tablas), ilustrando lo que habría que hacer en un examen de “lápiz y papel”. Obviamente, Matlab lo calcularía en una línea con `dsolve` o con `ilaplace(FdT/s/s)`. Sale la misma gráfica que la que sacaba el Control Systems Toolbox, obviamente, vídeo [105].

El diseño de un tratamiento térmico que no supere 50 grados de diferencia de temperaturas se aborda a partir del minuto [08:00]. La idea básica es, realmente, la de que si la pendiente es la mitad, la diferencia de temperatura también será la mitad... o sea, que lo resolveremos con proporcionalidades (regla de tres) de modo parecido a lo que se hizo en el vídeo [152].

Al final, con la rampa $T_{ext}(t) = 50/110 \cdot t$ no se supera el límite de temperatura, por lo que se propone un tratamiento térmico de crecer temperatura hasta 600 grados con 50/110 grados por segundo, y luego mantener el horno a 600 grados para terminar el tratamiento térmico el tiempo que sea necesario.

A partir del minuto [13:00] se plantea una modificación (hecha “a ojo”, sin cálculos concretos), opcionalmente podéis dejarlo aquí, pero si os interesa, se simula la introducción de la pieza en un horno precalentado a 50 grados, calentado a 50/110 grados por segundo hasta 680 grados y luego enfriado rápidamente a 600 grados. Ello consigue llevar la temperatura T_1 a su valor deseado de 600 grados, de forma un poc más rápida que el primero de los perfiles de temperatura propuestos.

4.6.3 Caso de estudio: transformador eléctrico

[180: tni1]

Modelado de un transformador eléctrico no ideal: ecuaciones físicas

Materiales: [modeltrafo.zip]


*** ☰ 09:37

[YouTube ▶]

Este vídeo discute el modelado de un transformador eléctrico no ideal (acoplamiento magnético menor a la unidad, pérdidas de Foucault) conectado a una carga resistiva. Aunque sea no *ideal*, las ecuaciones consideradas serán, no obstante, lineales (sin saturación magnética).

Contiene un repaso teórico muy breve a las ecuaciones de la física relevantes (acoplamiento magnético, autoinducción), así como código Matlab para obtener el modelo, usando la Symbolic Toolbox de Matlab. El resultado es un conjunto de ecuaciones que serán transformadas a función de transferencia en el vídeo [181] y a representación interna en el vídeo [182].

Aunque el título del material sea transformador “no ideal”, existen otros efectos (histéresis, capacidades parásitas, etc.) que hacen que los modelos de transformadores puedan complicarse todavía más, si se desea analizar su efecto con armónicos de altas frecuencias (ver, p.ej.: <http://hdl.handle.net/2117/112204>).

[181: tni2]	Representación transformada de Laplace (func. de transferencia) y respuesta temporal de transformador eléctrico no ideal	***  06:16
	Materiales: [modeltrafo.zip]	[YouTube ►]


Este vídeo continua el caso de estudio de un transformador eléctrico (no ideal, acoplamiento magnético incompleto) con carga resistiva cuyas ecuaciones de la física fueron discutidas en el vídeo [180], así como introducidas en Matlab mediante la Symbolic Toolbox.

Aquí se usa la Symbolic Toolbox para manipular las ecuaciones, sustituyendo las derivadas de intensidades I_1 , I_2 por su transformada de Laplace sI_1 , sI_2 con el comando **subs**.

Luego, se obtiene la representación en función de transferencia despejando las salidas de interés mediante el comando **solve**. Como alternativa, la representación en variables de estado se obtiene en el vídeo [182] a partir de las mismas ecuaciones de la física.

La parte final del vídeo obtiene la respuesta temporal ante corriente alterna (cond. iniciales nulas) mediante transformada de Laplace de la entrada (**laplace**), multiplicación por la función de transferencia, e inversa de Laplace (**ilaplace**).

Más ejemplos académicos (sin modelo físico subyacente) de obtención de respuesta temporal por fracciones simples y transformada inversa aparecen en los vídeos [111], [112], [113], [114].

[182: tni3]	Modelado de un transformador eléctrico no ideal: representación en variables de estado	***  07:45
	Materiales: [modeltrafo.zip]	[YouTube ►]

Este vídeo modela un transformador eléctrico (no ideal, acoplamiento magnético incompleto) con carga resistiva. Las ecuaciones de la física y su introducción en Matlab se discuten en el vídeo [180], que debería ser visionado con anterioridad a éste, y el análisis en transformada de Laplace se aborda en el vídeo [181].

En este vídeo, se usa la Symbolic Toolbox para manipular las ecuaciones de modo que se despejan las derivadas de las intensidades en primario y secundario del transformador (que son variables de estado, dado que aparecen bajo el signo de la derivada en las ecuaciones físicas), mediante el comando **solve**.

Obtiene las matrices (A, B, C, D) de la representación interna normalizada con el comando **jacobian**, escogiendo por ejemplo i_1 , i_2 y la tensión en la carga como salidas de interés (podrían escogerse otras) para formar la ecuación de salida y obtener las matrices C y D adecuadas.

Se comprueba la estabilidad y duración del transitorio mediante los valores propios (**eig**) de la matriz A .

Por último, se genera un objeto **ss** de la Control System Toolbox que encapsula la representación interna y se calcula la respuesta ante escalón (**step**), obteniéndose pues los transitorios en corriente continua.

Nota: el estudio del régimen permanente senoidal (corriente alterna) de este transformador se aborda en el vídeo [194].

El estudio del régimen senoidal (corriente alterna) de este transformador se aborda en el vídeo [tni4(09:21)] en la Sección 5.1.

4.6.4 Caso de estudio: calentador tubular

Una primera aproximación al modelado se vio en la sección 1.2.4.

[183: term1ePtf] Modelado dinámico de tanque de calentamiento como sistema de primer orden: linealización y Función de Transferencia (2 perfiles diferentes)

Materiales: [HeatExch1elementModeladoPadeyTF.pdf]

*** 12:58

[YouTube ►]

Este vídeo obtiene la representación en función de transferencia (bueno, *matriz* de transferencia, dado que son tres entradas las que se consideran) de los modelos de primer orden de un tanque de calentamiento que se han desarrollado en detalle en los vídeos [27] (agitación perfecta) y [28] (agitación imperfecta, perfil lineal de temperaturas interiores). El modelo exacto en forma de ecuación diferencial en derivadas parciales se discute en el vídeo [79].

El modelo de agitación perfecta es también linealizado en el vídeo [94], pero sin pasar a función de transferencia, conservando la expresión en representación interna para simularla con `ode45` y compararla con la simulación no lineal en un ejemplo numérico concreto; su visualización es aconsejable.

En este vídeo no se realiza simulación, sino que simplemente se linealiza el producto de caudales por temperaturas, y se presenta el resultado en transformada de Laplace (función de transferencia) resultante, tanto para el modelo de agitación perfecta como el modelo de perfil interno lineal de temperatura.

En el caso de perfil interno lineal de temperaturas, el resultado es un modelo de fase no mínima, que, según se comprobará, será igual a cierta aproximación de Padé de la solución de las ecuaciones en derivadas parciales que se discute en otro material. Se comprueba que si el tubo es largo o está poco aislado, entonces el modelo de fase no mínima podría tener ganancia negativa respecto a la temperatura de entrada, lo que es claramente contrario a toda intuición física: el modelo “real” no es de primer orden y todo esto son aproximaciones válidas en determinadas circunstancias y ante entradas “lentas” (de ancho de banda reducido).

La función de transferencia con perfil exponencial de temperaturas internas (vídeo [29]) tendrá los casos aquí discutidos como casos particulares. Se aconseja visualizar el vídeo mencionado sobre dicho caso exponencial.

[184: termedpsol] Ecs. derivadas parciales de calentador tubular de líquido: solución EDP con método de Laplace

Materiales: [HeatExchModelEDPsolucionFlujoCte.pdf]

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)


Este vídeo obtiene la solución de las ecuaciones diferenciales en derivadas parciales que rigen un intercambiador de calor tubular con una resistencia calefactora en su interior por el método de la transformada de Laplace sobre el componente “temporal” de la temperatura. Más detalles sobre el modelado EDP aparecen en el vídeo [79]. Dada la geometría tubular, el problema es unidimensional en el espacio. Las EDP son lineales sólo si el caudal que atraviesa el tubo es constante, por lo que así se asumirá.

Realmente, no obtenemos la “solución” propiamente dicha $T(x, t)$, sino la matriz de transferencia ante incrementos de potencia calefactora o incrementos de temperatura del fluido entrante. Soluciones numéricas y simulaciones serán el objetivo de otros materiales de la colección.

El resultado contiene términos derivados del retardo de transporte presente en los fenómenos físicos subyacentes, y tiene la forma:


$$T_{out}(s) = \frac{b \cdot (1 - e^{-\phi s} e^{-\phi a})}{s + a} Q(s) + e^{-\phi s} e^{-\phi a} T_{in}(s)$$

El caso donde sólo se considera el retardo se aborda (muy brevemente) en el vídeo [191]. La simulación de la respuesta ante escalón de estas funciones de transferencia se describe en el vídeo [185], continuación de éste. La comparación con aproximaciones de orden 1 en simulación se aborda en el vídeo [186].

[185: termexpstep] 09:58	Respuesta escalón de solución (función transferencia) a la EDP de un calentador tubular (Matlab)	*** 
Materiales: [CÓD.: HeatExchModelEDPsimula.mlx] [PDF]		[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo revisa la solución de la ecuación en derivadas parciales de un calentador tubular discutida en el vídeo [184], en el que se obtuvieron las funciones de transferencia ante incrementos de temperatura de entrada e incrementos de potencia calorífica. Luego, utilizando la control system toolbox, simula las respuestas ante escalón de temperatura de entrada y de potencia calorífica; se simulan un escenario con aislamiento térmico perfecto, y otro sin tal aislamiento.

[186: termlevsedp] 16:55	Comparación entre solución EDP y diferentes aproximaciones de orden 1 y 3 de un calentador tubular	**** 
Materiales: [CÓD.: HeatExchModelEDPsimulayOrdenReducido.mlx] [PDF]		[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo compara las soluciones exactas de la ecuación en derivadas parciales de un calentador tubular (detalle modelado en vídeo [79]) obtenidas en el vídeo [184] con aproximaciones de orden finito de las mismas.

Errata: existe un error en el vídeo en la línea 5 de código donde se propaga un factor de conversión de W a kW erróneamente. El MLX y PDF han sido corregidos, cambiando la constante de transmisión de calor \bar{k} para que los resultados numéricos del vídeo sigan siendo correctos.

Una vez aclarada la errata, continuemos con la descripción.

En concreto, mediante una aproximación de la exponencial que aparece en las funciones de transferencia en función de un cierto parámetro β , se derivan unos modelos de primer orden aproximados que, curiosamente, coinciden con los modelos obtenidos en el vídeo [29] a partir de un razonamiento físico. Esto es importante, porque los modelos físicos podrán ser simulados ante incrementos de caudal de entrada, mientras que las funciones de transferencia suponían caudal constante.

Los citados modelos físicos, a su vez, tenían como casos particulares a los modelos de temperatura interior homogénea (vídeo [27]) y los de perfil longitudinal lineal de temperatura (vídeo [28]).


También se comparan los modelos obtenidos usando una aproximación de Padé del retardo de orden 1 y de orden 3.

Las comparaciones se realizan tanto en ganancia estática como en transitorio ante escalones de temperatura de entrada y de potencia calorífica.

Conectando varias de estas aproximaciones podrían construirse aproximaciones del intercambiador mediante *elementos finitos*, que se discute en otros materiales de la colección, en concreto el vídeo [187] y simulaciones enlazadas en su descripción.

Centrándose únicamente en el fenómeno del “retardo de transporte”, podría ser de interés visualizar el vídeo [191] comparando diferentes aproximaciones del retardo.

[187: tubulFE1] Calentador/intercambiador tubular: modelado elementos finitos (I)

*****  14:47

Materiales: [[CÓD.:](#) HeatExchModelElemDiscre.mlx] [[PDF](#)]

[\[YouTube ►\]](#)

*[LINK TO ENGLISH VERSION](#)

Este video aborda el modelado mediante elementos finitos del intercambiador/calentador tubular discutido en los vídeos que comienzan con el [27] y continúan en los enlaces/referencias en la descripción del mismo. Otro ejemplo de modelado de elementos finitos en esta colección se aborda en el vídeo [278], que describe vibraciones longitudinales en un muelle.


Los primeros cinco minutos y medio revisan los muchos modelos discutidos en este caso de estudio, que van desde:

- Modelo como ecuación en derivadas parciales, EDP, [79]
- Modelos estáticos (estacionarios) derivados de dicha EDP (muy frecuentes en la práctica)
- Modelos de primer orden basados en balance de energía (ver vídeo [29], que es el caso general que engloba los de los vídeos [27] y [28]).

Esta revisión puede ser de interés en sí misma, para comprender/unificar los conceptos de todos esos vídeos previos.

A continuación, se plantea subdividir el intercambiador en N trozos iguales (elementos finitos) y cómo interconectar los elementos, tanto en diagrama de bloques como con las ecuaciones de estado y de salida de cada uno de los elementos (que se supondrán de primer orden, como arriba mencionado), de modo que formen un modelo de orden N que aproxime la solución “exacta” de la EDP (que sería el límite al hacer los elementos infinitesimales). Obviamente, el formar las N ecuaciones de estado con N muy grande se hace mediante un bucle y la Symbolic Math Toolbox, y explicar el código que lo realiza es el objetivo de la parte final del vídeo.

[188: tubulFEsim1] Calentador tubular, elementos finitos: simulación numérica (1)

*****  14:39

Materiales: [[CÓD.:](#) HeatExchModelElemDiscre.mlx] [[PDF](#)]

[\[YouTube ►\]](#)

*[LINK TO ENGLISH VERSION](#)

Este vídeo es continuación del vídeo [187] donde se indicaba cómo introducir en Matlab un modelo (simbólico) de elementos finitos de un calentador tubular, objeto de un caso de estudio a lo largo de múltiples vídeos.

En este vídeo, aparte de resumir brevemente el vídeo referido en los tres primeros minutos, se plantea cómo traducir de simbólico a numérico el problema, cómo definir las entradas y cómo integrarlo con un integrador numérico. Se comparan los tiempos de ejecución de `ode15s` y `ode45`, resultando ventajoso el primero de ellos en este caso cuando el número de elementos es grande; estos dos integradores numéricos también se comparan en el vídeo [39], dado que no siempre `ode15s` es mejor.

Se observan las simulaciones y parecen en general satisfactorias, aunque existen ciertos artefactos (oscilaciones de alta frecuencia durante el tiempo de retardo) al utilizar el coeficiente $\beta \approx 0.5$ en el modelo de cada elemento, relacionado con la aproximación de Padé del retardo en cada uno de ellos.

El detalle sobre esas oscilaciones, y la influencia del número de elementos y el valor de β sobre el resultado de la simulación se abordan en el vídeo [189], continuación de éste. También resultaría de interés visualizar el vídeo [191] sobre aproximación del retardo (fuera del caso de estudio) porque sus conclusiones aplican aquí y son similares a las obtenidas en el vídeo [189]. Una recapitulación y conclusiones finales sobre el modelado en este caso de estudio en general se abordarán en el vídeo [190].

[189: tubulFEsim2] Calentador tubular, elementos finitos: simulación numérica (2)
11:38

***** 

Materiales: [[CÓD.:](#) HeatExchModelElemDiscre.mlx] [[PDF](#)]

[[YouTube](#) ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo, continuación del [188] presenta simulaciones con distinto número de elementos y con distinto valor del coeficiente β diferente del teóricamente necesario para replicar con exactitud el balance de energía en régimen estacionario (equilibrio). Los cambios de β permiten disminuir los artefactos ante escalón de la simulación... aunque si las entradas no cambian de repente, pueden no ser un problema: la discusión es paralela al vídeo [191] donde se discute la aproximación del retardo de transporte específicamente.

Con esto se finaliza el caso de estudio sobre el calentador tubular que comenzó en el vídeo [27]; una recapitulación y conclusiones finales sobre el caso de estudio en general se abordarán en el vídeo [190].

[190: tubulconc] Calentador tubular: recapitulación y conclusiones sobre
complejidad de modelos

***  14:55

Materiales: [IntercConclusiones.pdf]

[[YouTube](#) ►]

* [LINK TO ENGLISH VERSION](#)


Este video es, en cierto modo, la conclusión del caso de estudio del intercambiador/calentador tubular discutido en los vídeos que comienzan con el [27] y continúan en los enlaces/referencias en la descripción del mismo.

Hace una recapitulación sobre las muchas formas que hay de dar un modelo, desde puramente “estático” hasta ecuaciones en derivadas parciales no lineales, pasando por modelos de primer orden basado en balance de energía de la temperatura *media*.

La complejidad excesiva de modelo da problemas de coste computacional, y posibles controladores complejos. Aparte, eso nunca solucionará problemas de dinámica no modelada (conducción en la tubería metálica, turbulencia) o errores paramétricos en coeficientes de transmisión de calor, o limitaciones de ancho de banda de la instrumentación de una aplicación concreta. Por ello, dependiendo de la relación entre el tiempo de residencia del fluido y el tiempo de establecimiento que se busca en cada aplicación concreta, la mejor elección de ingeniería será una cierta complejidad “razonable” del modelo.

4.6.5 Sistemas con retardo

[191: aproxret] Aproximación del retardo: simulación y comparación de
opciones (PDE, Padé, elementos finitos)

***  15:59

Materiales: [[CÓD.:](#) aproxreteltemaPADE2.mlx] [[PDF](#)]

[[YouTube](#) ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute qué es el retardo de transporte, con función de transferencia $\exp(-ds)$, y plantea tres aproximaciones del mismo a un orden finito N :

- Aproximación de Padé de orden N
- Aproximación de N elementos finitos de primer orden (cada uno de ellos con Padé de orden 1)
- Aproximación de N elementos finitos con sólo polos.

La parte inicial revisa la EDP de transporte y su solución por Laplace (muy rápido, no es objetivo principal, desarrollo similar al vídeo [184]). Puedes saltártela y adelantar al minuto 3 si no te interesa la EDP de transporte, no pasa nada.

El vídeo compara las respuestas temporal (ante escalón, rampa y filtro de 1er orden) y en frecuencia con distintos valores de N y saca conclusiones acerca de las condiciones de validez de las aproximaciones.

Las aproximaciones de elementos finitos del retardo de transporte son casos particulares de un modelo más complejo con pérdidas de calor y resistencias de un calentador tubular, cuya ecuación en derivadas parciales se aborda en el vídeo [79], y cuya aproximación por elementos finitos se discute en el vídeo [187].

La aproximación de términos de serie de Taylor entre dos expresiones, que es la base de la aproximación de Padé, también se discute en el vídeo [289] para aproximar sistemas de orden superior a ‘primer orden con retardo’.

Capítulo 5

Respuesta en frecuencia de sistemas lineales

5.1 Fórmulas y diagramas de respuesta en frecuencia

La salida de un sistema lineal ante una entrada senoidal es una senoide de la misma frecuencia, pero con diferente amplitud y desfase respecto a la senoide de entrada. Los siguientes videos abordan en el caso monovariable su cálculo y representación en diagrama de Bode.

[192: freqresp]	Respuesta en frecuencia de sistemas dinámicos lineales en función de transferencia	**  10:57
	Materiales: [FreqRespH.pdf]	[YouTube ►]

En este vídeo:

- Se motiva (vibraciones, corriente alterna, control, filtrado) la necesidad de la respuesta en frecuencia.
- Se demuestra mediante la descomposición en fracciones simples, y transformada de Laplace inversa, la fórmula del régimen estacionario senoidal de un sistema de función de transferencia $G(s)$, de una entrada y una salida.
- Se calcula, como ejemplo, la respuesta en frecuencia de $G(s) = 1/(s + 1)$ ante entrada $u(t) = \cos(2t)$, utilizando el comando **freqresp**.

Nota: El cálculo de régimen estacionario senoidal en un sistema en representación interna se discute en el vídeo [165]. Obviamente, produce resultados equivalentes.

[193: bode]	Respuesta en frecuencia: Diagrama de Bode	**  09:49
	Materiales: [BodeFiltros3H.pdf]	[YouTube ►]

Este vídeo presenta el diagrama de Bode de amplitud y desfase. Los contenidos básicos de respuesta en frecuencia (video [192]) se suponen conocidos.

- Se explican las escalas logarítmicas en frecuencia (décadas) y amplitud (decibelios).
- Se esboza brevemente la idea de que en escala logarítmica los diagramas pueden ser representados como fragmentos de recta, a partir del ejemplo de un filtro paso-banda $G(s) = 10s/(s + 1)/(s + 1000)$.
- Asimismo, la multiplicación de funciones de transferencia (conexión en cascada) resulta en la suma de los diagramas de Bode, debido al uso (intencionado para conseguir precisamente esto) de escalas logarítmicas.

nota: Muchos textos y vídeos en la red presentan con más detalle el trazado de los diagramas de Bode de forma manual. Estos contenidos son “clásicos” en cursos iniciales de teoría de control, dado que existen metodologías de diseño de controladores basados en dichos diagramas; también son de interés en diseño de filtros. Su visualización es recomendable para afianzar las ideas presentadas aquí, aunque no imprescindible, dado que Matlab los dibuja fácilmente con el comando `bode`.

Ejemplos adicionales en esta colección de vídeos sobre el tema aparecen en el vídeo [195] (filtros) y ejemplos aplicados a un fragmento musical en los vídeos [196] y [197].

Los vídeos <https://www.youtube.com/watch?v=6Rp-J8T-mjM&list=PL2A3EDF6FCA7A27E6&index=41&t=0s> y <https://www.youtube.com/watch?v=aPmY6SJphN4&list=PL2A3EDF6FCA7A27E6&index=42&t=0s> del canal [controltheoryorg](https://www.youtube.com/channel/UCv3p3v3p3p3p3p3p3p3p3p3) presentan con más detalle el trazado de los diagramas de Bode de forma manual. Estos contenidos son “clásicos” en cursos iniciales de teoría de control, dado que existen metodologías de diseño de controladores basados en dichos diagramas; también son de interés en diseño de filtros.

Ejemplo:

[194: tni4] Análisis en frecuencia de transformador eléctrico no ideal:
régimen estacionario, impedancia, simplificación de polos a
altas frecuencias

Materiales:

**** 09:21

[YouTube ▶]

[modeltrafo.zip]

Este vídeo concluye el caso de estudio sobre un transformador eléctrico (no ideal, acoplamiento magnético incompleto) con carga resistiva.

En este vídeo se calcula la respuesta en frecuencia (diagrama de Bode, detalles sobre él en [193]) de la tensión en la carga (secundario) ante la entrada de tensión en el primario (teoría sobre respuesta en frecuencia, en vídeo [192]). Se comprueba que el diagrama tiene forma de filtro paso-banda: existe un intervalo de frecuencias donde la relación de transformación es la calculada, pero a frecuencias muy bajas la tensión en secundario cae (los transformadores no funcionan en corriente continua) y lo mismo ocurre a frecuencias altas dado que las impedancias debidas a las autoinducciones crecen.


El régimen estacionario senoidal a 50 Hz (100π rad/s) se calcula en amplitud y fase usando el comando `freqresp`.

El vídeo presenta la diferencia entre el transitorio (que sólo dura 2 ms, que se calculó mediante Transformada de Laplace en el vídeo [181]) y el régimen estacionario senoidal.

Por último, se relacionan los resultados con lo que podría haberse calculado mediante fórmulas de impedancia (ingeniería eléctrica, corriente alterna). También se reduce el modelo a un modelo de primer orden eliminando el polo rápido (con frecuencia de corte de $> 1500\text{rad/s}$, esto es lo que se denomina reducción *modal*, estudiada en profundidad en el vídeo [294]). Se comparan las respuestas temporal y en frecuencia del modelo original y el reducido.

Caso multivariable: Aunque en estos momentos no sea necesario abordarlo, para limitar la complejidad de la exposición por obvias razones didácticas, el concepto de respuesta en frecuencia multivariable es abordado en el vídeo [propsresto(12:48)], y el vídeo [freq2mass(07:01)]. Requiere, como conocimiento previo, el concepto de valores singulares (Sección A.4, vídeos [svdOA(11:35)] y [svd2(17:10)]).

Nota: El concepto de respuesta en frecuencia puede ser utilizado para hacer análisis de estabilidad en bucles de control, dando lugar a lo que se conoce como criterio de Nyquist (años 1930), discutido en la Sección 10.8.1. Como la respuesta en frecuencia puede ser determinada experimentalmente mediante un ensayo ante

entrada senoidal, ello permite determinar la estabilidad de bucles cerrados de control a partir de un ensayo experimental en bucle abierto sin conocer con precisión los polos y ceros de los elementos intervinientes, con una metodología puramente “gráfica”, ventajosa antes de la generalización de los computadores en la década de 1980. *** 

5.2 Filtros analógicos elementales

[195: filt] Respuesta en frecuencia: filtros sencillos en tiempo continuo
Materiales: [BodeFiltros3H.pdf]

**  09:29
 [YouTube ▶]

Este vídeo, continuación del vídeo [193], presenta los componentes básicos de filtros, tanto con polos reales como en el caso de polos complejos (resonantes).

- se revisan las escalas del diagrama de Bode.
- Se calcula la respuesta de $(s + a)$ a frecuencia baja $\approx a$, y a frecuencia alta $\approx j\omega$.
- Se calcula el diagrama de Bode del filtro paso bajo $K/(s + a)$.
- Se calcula el diagrama de Bode del filtro paso alto $Ks/(s + a)$.
- Se calcula el diagrama de Bode del filtro paso banda $Ks/(s + a)/(s + b)$.
- Se presenta en filtro resonante de segundo orden. Se pone como ejemplo un filtro ranura (notch) de banda muy estrecha.

Ejemplos Matlab del comportamiento de estos filtros en procesamiento de señal de audio aparecen en los vídeos [196] (no resonantes), y [197] (caso resonante).

Un ejemplo (excesivamente complejo en este momento si eres un estudiante que comienza en esta materia) de comportamiento paso-banda en control de procesos (tanque de mezclado) se discute en el vídeo [543]. ****

5.2.1 Ejemplos

Los siguientes videos presentan la aplicación de filtros sencillos continuos a un fragmento musical:

[196: filtml1] Filtros analógicos paso alto/bajo/banda: ejemplo Matlab
 (control systems toolbox)
Materiales: [Cód.: filtestest.mlx] [PDF]

**  10:58
 [YouTube ▶]

Este vídeo discute íntegramente código Matlab (Control Systems Toolbox). En este video se aplican filtros (polos reales) de paso alto/bajo/banda (descripción teórica y diagramas de Bode en vídeo [195]) sobre un fragmento de música, para escuchar su efecto. Se usa la control system toolbox. La suma con coeficientes a elección del usuario de dichos filtros constituiría un ecualizador gráfico de 3 bandas.

Filtros resonantes sobre fragmentos musicales se abordan en el vídeo [197].

[197: filtmlre] Filtros resonantes: ejemplo Matlab (control systems toolbox)
Materiales: [Cód.: filtestestresonant.mlx] [PDF]

***  06:46
 [YouTube ▶]

Este vídeo presenta un ejemplo de Matlab (Control Systems Toolbox) que aplica un filtro resonante tipo ranura (*notch*) para eliminar únicamente una banda muy estrecha alrededor de la frecuencia de una nota concreta de una escala musical. Un segundo ejemplo de filtro deja pasar sólo una nota de la escala.

Los tipos de filtro se basan en la expresión:

$$G(s) = \frac{s^2 + 2\psi_N\omega s + \omega^2}{s^2 + 2\psi_D\omega s + \omega^2}$$

donde si $\psi_N \rightarrow 0$, $\psi_D > \psi_N$ se obtiene un filtro ranura, y si $\psi_D \rightarrow 0$, $\psi_N > \psi_D$ entonces se obtiene un filtro resonante que sólo deja pasar una banda muy estrecha de frecuencias cercana a ω .

Filtros *no* resonantes en fragmento musical se abordan en el vídeo [196].

Nota: Las fórmulas de progresiones geométricas para cálculo de las frecuencias del armónico fundamental de las notas de una escala (temperada) no son, obviamente, objetivo de la disciplina de Ingeniería de Sistemas y Automática. La explicación de esa parte del código puede consultarse en fuentes de Teoría de la Música o Acústica o, por ejemplo, <http://elclubdelautodidacta.es/wp/2012/08/calculo-de-la-frecuencia-de-nuestras-notas-musicales/>.

Nota: El diseño riguroso de filtros para procesamiento de audio no es objetivo de este material. Hay otros filtros continuos y digitales más sofisticados para este tipo de aplicaciones, bien estudiados en el área de Ingeniería de Telecomunicaciones y el Signal Processing Toolbox de Matlab.

5.3 Transformada de Fourier

Aunque la respuesta temporal y la respuesta en frecuencia considerada hasta este momento utiliza la transformada de Laplace, determinadas aplicaciones de filtrado/suavizado no causal (con datos “grabados” de modo que se conocen valores futuros de una señal) se formalizan teóricamente mediante la transformada de Fourier, a la que se dedica esta sección. La transformada de Fourier también se utiliza en el análisis de contenido en frecuencia de “ruidos coloreados”.

Nota: Los contenidos teóricos de los materiales de esta sección no son estrictamente necesarios en un primer momento y esta sección puede ser fácilmente saltada hasta que el alumno considere conveniente profundizar en el filtrado no causal o en los procesos con ruido. Una presentación “rápida” e intuitiva de un filtro no causal más sencilla puede encontrarse en el vídeo [cau(11:29)], o en la documentación del comando `filtfilt` de Matlab.

[198: tfour]

Transformada de Fourier: definiciones básicas, interpretación,
relación con Laplace

Materiales:

**** ☰ 19:27

[YouTube ▶]

[FourierTransform.pdf]

Este vídeo revisa (se supone conocido de otros materiales) el concepto de “serie de Fourier” para señales periódicas (y de sus “armónicos” relacionados) que, cuando el período tiende a infinito, motiva la definición de la “transformada de Fourier” para representar señales arbitrarias en el dominio de la frecuencia.

La transformada está definida para señales con $t \in \mathbb{R}$, lo cual difiere de la transformada de Laplace “causal” (unilateral) que considera señales con $t \in \mathbb{R}^+$ (origen de tiempos en $t = 0$). Para que la transformada de Fourier converja y esté acotada, las señales sobre las que se aplica deben ser no persistentes en ambas direcciones temporales (tender a cero tanto cuando $t \rightarrow -\infty$ como cuando $t \rightarrow +\infty$).


El vídeo analiza la relación con la transformada de Laplace clásica (unilateral), y con la transformada de Laplace bilateral que considera señales que pueden tener valores no nulos antes de $t = 0$.

Se plantea un ejemplo de transformada de Fourier de una exponencial. Se enfatiza que lo que en Laplace unilateral sería “inestable” en transformada de Fourier es “anticausal”.

Por último, el vídeo enuncia rápidamente, sin demostración, propiedades de utilidad de la transformada de Fourier para aplicaciones (linealidad, retardo, derivada, Parseval), similares a las de la transformada de Laplace.

[199:tfourm] Transformada de Fourier, filtrado no causal: ejemplo Matlab

Materiales: [[CÓD.:](#) ejemploFourier.mlx] [[PDF](#)]

****  16:01

[YouTube ▶]

Este vídeo resuelve el ejemplo exponencial que se había planteado en el vídeo [198] mediante los comandos `fourier` y `ifourier` de la Symbolic Toolbox de Matlab, comprobando que efectivamente, produce los componentes causales y anticausales discutidos en la teoría.

La segunda parte del vídeo plantea la respuesta ante escalón unitario en $t = 0$ de:

1. un filtro $3/(s+3)$, que resulta idéntico al caso `step` o las operaciones con transformada de Laplace unilateral (tanto el filtro como el escalón son “causales”),
2. un filtro $3/(-s+3)$ que produce un filtro “anticausal” de modo que todo el transitorio se produce antes de $t = 0$: obviamente, esto sólo podría ser implementado en la realidad con una secuencia de datos “grabados” de modo que el filtro pudiera saber “el futuro”.
3. un filtro combinado $9/(9-s^2) = 9/(3+s)/(3-s)$ que tiene componentes causales y anticausales y cuyo diagrama de Bode tiene fase cero (transf. Fourier $9/(\omega^2+9)$ real). En este filtro, el momento en el que se produce el escalón está en el centro del transitorio.

La parte final del vídeo compara las manipulaciones “formales” con la transformada de Fourier en la Symbolic Toolbox con el resultado de la simulación numérica con el comando `filtfilt` del filtro $3/(s+3)$ hacia adelante en el tiempo (causal) y el resultado hacia atrás en el tiempo (anticausal). Se comprueba como la simulación coincide con la teoría, salvo errores en condiciones iniciales (la teoría requeriría simular “infinito tiempo”) y efectos de discretización (`filtfilt` necesita filtros discretos, que se han obtenido del continuo a partir de la transformación bilineal, usando `c2d`, con `c2d(G,Ts,'tustin')` siendo T_s suficientemente pequeño; los detalles sobre la discretización no son relevantes en este momento, y se discutirán en profundidad en otros materiales, a los que se remite al lector para la comprensión en profundidad del fragmento de código asociado).

Los filtros no causales en representación interna son abordados en el vídeo [607].

El filtrado (suavizado) y la separación señal/ruido tienen, obviamente, una interpretación estadística. La Transformada de Fourier es una herramienta importante en el procesamiento estadístico de señal, motivando lo que se denomina “densidad espectral de potencia” cuando se aplica a señales aleatorias. El paso de transformada determinista a señales aleatorias se aborda en la Sección 7.1.1.

Nota: Los vídeos anteriores discuten la transformada de Fourier en tiempo continuo (definida con integrales); la transformada de Fourier en tiempo discreto está definida con sumas en vez de integrales, no se discute aquí por brevedad, y se remite al lector a la abundante bibliografía sobre ella en textos de procesamiento digital de señal en la que dicha transformada tiene un papel clave.

Capítulo 6

Sistemas de tiempo discreto: representación y análisis

Dado que en control por computador se toman medidas de sensores y se calculan acciones de control a intervalos (usualmente regulares), en este capítulo se discuten cuestiones relativas a sistemas dinámicos que procesan “secuencias” (sucesiones) de señales, esto es, señales en tiempo *discreto*.

6.1 Respuesta temporal

En representación entrada-salida se utiliza la función de transferencia discreta y la transformada \mathcal{Z} . No dispongo de vídeos realizados por mí sobre el tema.

► *Introducción a la transformada \mathcal{Z} :*

[Andrés Camacho]

[Enlace]

6.2 Respuesta en frecuencia

[200: rfd] Respuesta en frecuencia de sistemas en tiempo discreto $G(z)$

Materiales: [FreqRespDISC.pdf]

***  12:45

[PLAY ►]

En este video se discute el cómputo de la respuesta en frecuencia de sistemas discretos, en concreto, se demuestra que la salida de un sistema con función de transferencia $G(z)$ ante $u(k) = \cos(\omega T k)$ es igual a $y(k) = M \cos(\omega T k + \varphi)$ donde M y φ son el módulo y argumento de $G(e^{j\omega T})$, respectivamente. Este vídeo es el “paralelo” en el caso discreto al vídeo [192] que abordaba el caso continuo.

La obtención de sistemas discretos a partir de la observación de sistemas continuos a intervalos regulares es parte de lo que se conoce como “discretización”. Este tema se aborda en profundidad en el Capítulo 11.


Capítulo 7

Procesado estadístico de señal: procesos estocásticos

Nota 1: Para la mejor comprensión de los resultados de este capítulo, es muy recomendable la revisión de los conceptos básicos de Estadística discutidos en el Apéndice B.


Nota 2: En el caso de señales aleatorias en tiempo discreto con dinámica lineal, muchos conceptos pueden ser abreviados/simplificados. Un lector “con prisas”, sin interés en el enfoque frecuencial/continuo, podría saltar a la Sección 7.2 en una primera lectura, y decidir posteriormente si la temática le interesa lo suficiente para empezar desde cero, con los fundamentos en la Sección 7.1.

7.1 Procesos estocásticos: definiciones básicas y ejemplos

[201: estoc1]	Procesos estocásticos (funciones aleatorias): motivación, definición y ejemplos para estudiantes ingeniería	**  15:47
	Materiales: [estocas1.pdf]	[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo plantea el concepto de “proceso estocástico”, que se utiliza en estadística, como formalización de lo que se entendería en ingeniería como una “señal con ruido”. Se discute su definición fundamental y ejemplos en procesamiento de señal industrial, audio-vídeo, vibraciones, así como los objetivos del estudio de los mismos. El objetivo del procesamiento de señal son los procesos estocásticos en el tiempo (series temporales) que los siguientes vídeos analizan; de todas formas, algunas de las ideas son generalizables de forma sencilla a procesos de dimensión mayor asociados a funciones aleatorias sobre un dominio espacial bidimensional (análisis de ruido en imágenes, por ejemplo) o incluso mixto (análisis de señales de vídeo); véase, por ejemplo, el caso bidimensional en el vídeo [250].

[202: estoc2]	Señales con ruido: procesos estocásticos en el tiempo (series temporales), definiciones básicas	***  11:00
	Materiales: [estocas2.pdf]	[YouTube ►]

Este vídeo presenta definiciones particulares a lo que se entiende como “serie temporal” (proceso estocástico indexado sobre el tiempo continuo o discreto). Introduce el concepto de realización. Las propiedades de la señal, como medias y varianzas, serían funciones del tiempo. También se define la covarianza o correlación entre variables aleatorias correspondientes a dos instantes de tiempo: la correlación estadística entre señales en distintos instantes permitirá extender el concepto de predicción lineal óptima del vídeo [967] a series temporales, que se explica más adelante en otros vídeos de esta colección. El vídeo

finaliza enumerando los posibles objetivos del análisis de este tipo de señales (identificación, predicción, control predictivo, filtrado, suavizado, interpolación) y resumiendo las ideas principales en sus conclusiones.

Nota: los procesos estocásticos pueden venir bien con matrices de varianzas-covarianzas, y correlaciones “caídas del cielo” (o computadas experimentalmente) con lo que ya se puede aplicar las fórmulas de predicción, o bien ser generados a partir de un modelo físico de un sistema con entradas tipo ruido (en tiempo discreto –ecuaciones en diferencias estocásticas, vídeo [219]–, o en tiempo continuo –ecuaciones diferenciales estocásticas, vídeos [223] y [226], entre otros–). No es objetivo de este vídeo entrar en el detalle de “de donde vienen”, sino de definir “qué son”.

[203: estoce] Procesos estocásticos estacionarios

Materiales: [estocs3stac.pdf]

****  13:12

[YouTube ►]

Este vídeo explica el concepto de estacionariedad en series temporales, que generaliza el concepto de “equilibrio” (valor constante) al concepto de “función de densidad conjunta que no depende del tiempo” (con más precisión, independiente de desplazamientos temporales) para definir los procesos estacionarios en sentido *estricto*, y una versión más relajada del mismo sólo en términos de media, varianzas y covarianzas constantes (estacionariedad en sentido *amplio* o *débil*) que resulta de gran utilidad en la práctica ya que las operaciones lineales sobre señales en muchos casos sólo requieren del cálculo de dichas medias, varianzas y covarianzas por lo que si son constantes los cálculos resultan más sencillos.

[204: estoctm] Procesos estocásticos en el tiempo (series temporales, introducción): ejemplo Matlab

Materiales: [[CÓD.:](#) estocastest1.mlx] [[PDF](#)]

****  09:58

[YouTube ►]


Este vídeo presenta un ejemplo Matlab donde se simula la repetición múltiples veces de un experimento de descarga de un condensador (simulación de sistema de primer orden sujeto a entradas con ruido) que se había esbozado y comentado en vídeos anteriores. Se calculan las medias y varianzas para cada instante, así como las covarianzas para cada par de instantes, y se observa que en las partes finales de las simulaciones se alcanza el estado estacionario (una vez desaparecen los efectos transitorios de condiciones iniciales).

Nota 1: En bastantes aplicaciones, “prolongar” el experimento aporta la misma información que “repetirlo”, esto será analizado teóricamente en el vídeo [205] y el código en ese caso será mucho más sencillo, como se describe en el vídeo [206].

Nota 2: En este vídeo se “simula” una ecuación ante ruido y se calculan las matrices de varianzas-covarianzas a partir de las realizaciones. En aplicaciones de *control*, las covarianzas se calculan a partir de las ecuaciones en diferencias o diferenciales que describen el modelo del proceso (ver vídeo [229], por ejemplo). En aplicaciones de *filtrado*, estas matrices se obtienen a veces de descripciones de filtros en el dominio de la frecuencia; entonces, se puede hacer el proceso inverso, calcular realizaciones a partir de la matriz de varianzas-covarianzas; aunque la teoría de los temas frecuenciales se discute en el vídeo [207], podría ser de todos modos de cierto interés visionar el vídeo [208] para acabar de comprender el concepto de proceso estocástico como colección de variables aleatorias.

[205: ergod] Procesos estacionarios ergódicos

Materiales: [estocs4ergo.pdf]

****  10:51

[YouTube ►]

En este vídeo se define e interpreta el concepto de proceso estocástico estacionario *ergódico*. Básicamente, son el tipo de procesos estacionarios en los que, como se repiten las características estadísticas idénticamente en todos los instantes de tiempo, los cálculos con “repeticiones del experimento” coinciden con los cálculos obtenidos de un único experimento (realización) “muy largo”, cuando el número de repeticiones (realizaciones) y la duración de la única realización considerada tienden a infinito.

La definición formal de ergodicidad y sus teoremas asociados no son sencillos, pero informalmente se requiere que no haya “varios puntos de equilibrio” o “varios atractores (conjuntos invariantes)” en la dinámica subyacente que genera la señal. Los procesos lineales estables sometidos a ruido blanco de distribución normal son ergódicos, siempre que tanto el sistema dinámico como la media y varianza del ruido que excita al sistema sean invariantes en el tiempo.

Un ejemplo Matlab de alguna de las ideas expuestas en este material aparece en el vídeo [206].

[206: ergodm]
Procesos estacionarios ergódicos: ejemplo Matlab serie temporal
**** 10:24

Materiales: [[CÓD.](#): estocas2ergodic.mlx] [[PDF](#)]
[YouTube ▶]

Este vídeo utiliza la suposición de ergodicidad para calcular la media, varianza y autocovarianza una vez se alcanza el estado estacionario a partir de una única serie temporal “larga” en vez de la repetición de experimentos abordada en el vídeo [204]. Se usan los comandos `mean`, `var` y `xcov` sobre dicha serie temporal.

En cuanto a la duración del experimento, es importante recalcar que cuando se calculan medias sobre una serie temporal donde las muestras consecutivas tienen correlación importante, entonces el número de muestras “efectivo” es menor: si la correlación es alta cuando la distancia temporal es pequeña, básicamente las muestras cercanas en el tiempo proveen copias de la misma información. Por ello, el experimento debe ser tanto más largo cuanto más largo sea el intervalo donde hay una correlación significativa. Grosso modo, el número de muestras efectivo para el cálculo de medias es el número de muestras tomado dividido por el área de la autocorrelación.

Nota: En procesamiento digital de señal, suponer ergodicidad suele ser el enfoque más común cuando se dispone de una señal estacionaria (no se observa ningún efecto de condiciones iniciales) y se asume que sólo existe un único atractor (punto de equilibrio) o se desea estudiar (aproximar) con técnicas lineales el comportamiento alrededor de ese único punto de equilibrio (por ejemplo, para hacer control lineal). En otras aplicaciones (condiciones iniciales específicas, variación temporal de la dinámica en un proceso por lotes, etc.) podría ser necesario “repetir” el experimento en vez de “prolongarlo”, analizándolo como en el vídeo [204], formalmente correcto en todos los casos.

7.1.1 Descripción frecuencial de los procesos estacionarios

Nota: Para la comprensión de esta sección, se recomienda revisar los conceptos básicos sobre la Transformada de Fourier de la Sección 5.3, al menos el material sobre la definición y propiedades [tfour(19:27)].

[207: psd]
Densidad espectral de potencia en señales aleatorias estacionarias
**** 17:24

Materiales: [estocs5PSD.pdf]
[YouTube ▶]

En un proceso estacionario, la señal temporal $x(t)$ es aleatoria, pero la media, varianza y covarianza entre instantes de tiempo, si se repitiera dicho experimento infinitas veces, sí serían constantes. Obviamente, si $x(t)$ es aleatorio, también lo será su contenido en

frecuencia. Sin embargo, en muchas señales físicas se “ven” oscilaciones (correlación entre los valores en un instante y los valores cierto tiempo “período” después). La relación entre la “correlación entre dos instantes separados” y “contenido frecuencial” es lo que explora este vídeo.

En concreto, este vídeo presenta el concepto de “*densidad espectral de potencia*” (power spectral density, PSD, en la literatura en lengua inglesa).

En los primeros cinco minutos de vídeo, se revisan los conceptos de proceso estacionario y de transformada de Fourier. La segunda mitad del vídeo esboza la demostración del teorema de Wiener–Khinchin–Einstein, que relaciona la respuesta “temporal” (covarianza entre diferentes instantes de tiempo) con la respuesta “frecuencial” (densidad espectral de potencia).

La identidad de Parseval, obtenida de la fórmula de la transformada inversa de Fourier, indica que la varianza de una señal es la integral de su PSD sobre el eje de frecuencias. Ello posibilita la interpretación de la PSD como “contribución de los componentes de frecuencia f a la varianza total”, o a la integral en un intervalo de frecuencias como “varianza en dicho rango de frecuencia”. La dualidad entre la interpretación estadística y la interpretación frecuencial permite entender determinados problemas de estimación en series temporales como problemas de *filtrado* en el dominio de la frecuencia. Se aconseja visualizar el vídeo [241] para ver ejemplos Matlab de pares autocovarianza/densidad espectral y su uso en filtrado.

7.1.2 Generación de muestras de un proceso estocástico gaussiano

[208: dadogaus] [Generación de trayectorias de un proceso estocástico gaussiano de autocovarianza dada](#)

Materiales: [[CÓD.:](#) TirarDadoProcesoGaussiano.mlx] [[PDF](#)]

**** 12:46

[YouTube ▶]

En este vídeo se generan trayectorias de un proceso estocástico gaussiano con función de covarianza dada $\kappa(x_1, x_2)$. Una vez se tiene el rango de x a “simular” (discretizado a N puntos, para que el número de variables aleatorias sea finito, claro) lo único que hay que hacer es aplicar las ideas basadas en diagonalización del vídeo [956] cuya visualización previa a éste se aconseja a la matriz $N \times N$ obtenida calculando las covarianzas dos a dos. Como alternativa computacional más rápida, también se calcula aquí la factorización de Cholesky de la matriz de varianzas.

En este vídeo se ilustran realizaciones de el régimen estacionario de un filtro de primer orden y un filtro de segundo orden, usando la dualidad entre respuesta en frecuencia (power spectral density) y función de autocovarianza: la respuesta en frecuencia es la transformada de Fourier de la autocovarianza (ver vídeo [207] con detalle y demostración asociada); esto se hace por comprender mejor dicha interpretación, pero realmente si no se hace uno preguntas sobre el origen de $\kappa(x_1, x_2)$ sólo hay que directamente aplicar los resultados del vídeo [956] antes mencionado.

La parte final del vídeo comprueba empíricamente que la varianza y covarianza entre muestras de las realizaciones generadas coincide, efectivamente, con aquélla especificada por la función κ (cuando el número de muestras tiende a infinito).

[209: gpsamb] [Realizaciones de proceso gaussiano \(ejemplo Matlab\)](#)

Materiales: [[CÓD.:](#) GPSSampletests2ESP.mlx] [[PDF](#)]

*** 13:32

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute cómo calcular ‘realizaciones’ (funciones al azar $f(x)$) de un proceso estocástico gaussiano de media $\bar{f}(x)$ y covarianza $k(x_1, x_2)$ dada. Como el proceso es de ‘tiempo continuo’, sólo se generarán muestras del mismo en un conjunto finito de puntos de prueba.


Se discute el código para generar la matriz de covarianzas en dicho conjunto de puntos de prueba, el significado del kernel de covarianza estacionario elegido (exponencial cuadrático, pero puede ser cualquier otro), la estructura tipo banda de la matriz de covarianza y se utiliza `mvnrnd` para generar las realizaciones del proceso. El detalle interno del comando `mvnrnd` requiere hacer, por ejemplo, diagonalización o descomposición de Cholesky de la matriz de covarianza, ver vídeos [956] o [208] para más información.

Para entender mejor este tipo de procesos, se cambia el parámetro de distancia de correlación en abscisa de la función de covarianza, se añade ruido blanco de medida (incrementando la diagonal de la covarianza), etc.

El vídeo [210], continuación del presente vídeo, discutirá el mismo problema (generar realizaciones) cuando se dispone de observaciones en algunos puntos de ‘medida’ del valor del proceso (realizaciones del *posterior*).

[210: gpsambpo] Realizaciones de proceso gaussiano con observaciones (muestras del posterior): ejemplo Matlab

Materiales: [[CÓD.](#): GPSampletests2ESP.mlx] [[PDF](#)]

****  14:53

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute cómo calcular realizaciones de un proceso estocástico cuando se tienen observaciones de su valor en un conjunto de puntos. Por ejemplo, el proceso es la velocidad en distintos puntos de una autopista y se desea ‘interpolar’ para un coche concreto las trayectorias más probables sabiendo que en dos posiciones donde había un radar su velocidad era de 110 y de 80 Km/h.

La primera parte del vídeo revisa conceptos previos y la mejor predicción lineal, y discute el código para generar media y matriz de covarianzas a posteriori en un conjunto de puntos de test.

La segunda parte del vídeo dibuja los intervalos de confianza de ese posterior y genera realizaciones varias (funciones aleatorias) para ilustrar el concepto, usando `mvnrnd`. El desarrollo es bastante paralelo al vídeo [209], cuya visualización previa se recomienda.

La parte final del vídeo presenta simulaciones del significado de cambiar la distancia de correlación en el kernel de covarianza.

7.1.3 Procesos estocásticos de varias salidas (MIMO)

[211: mimoGP1EN] (1/5) Multi-Output Gaussian Processes: Motivation

15:34

Materiales: [[CÓD.](#): GPmultioutputpart1.mlx] [[PDF](#)]

*** 

[YouTube ►]

This video introduces and motivates multi-output stochastic processes, based on two basic examples:

- Temperature and humidity readings at weather stations in different positions, from which you want to ‘interpolate’
- The statistical relationship between the position and velocity of a mass subject to random accelerations.

In these examples, we will have a variable, say $y_1(x)$, and another one, $y_2(x)$, so that apart from the ‘marginal autocovariances’, say, $E[y_1(x_a)y_1(x_b)]$, $E[y_2(x_a)y_2(x_b)]$ (we assume zero mean for notational simplicity), we could have a covariance between the signals $E[y_1(x_a)y_2(x_b)]$ that would allow us to interpolate humidity with temperature data or estimate speed with position data, for example.

How are these covariances generated? Well, the theory of stochastic differential equations and filtering justifies the second case (position, velocity)... The former case (temperature, humidity) could be more ‘descriptive’ or assuming certain components (underlying latent variables) to be present in the data-generating mechanism. The ‘intuitive’ details of this case are discussed in the video [212], a continuation of this one. Stochastic differential equations are a topic of greater complexity (but of greater theoretical interest, too), discussed in other materials.

[212: mimogp2EN] (2/5) Multi-Output Gaussian Processes: representation; code
21:58 for realizations

Materiales: [[CÓD.:](#) GPmultioutputpart1.mlx] [[PDF](#)]

[[YouTube](#) ►]

This video, a continuation of the video [211], discusses the representation of multi-output stochastic processes in two ways:

- as a process that, indeed, has an auto-covariance function $k(x_1, x_2)$ whose result is a matrix of size according to the number of outputs
- as well as a stochastic process with one more input, $k(\xi_1, \xi_2)$, with $\xi = (x, j)$ being an augmented input, a pair of abscissa + index (say, 1 or 2 if we are in a two-output case) indicating which output we are referring to, henceforth with $k(\xi_1, \xi_2)$ being a scalar.

Actually, both representations are equivalent, as one would intuitively expect, and it is just a matter of preference when writing code in which one is used.

Although it is not the only option, generating these multi-output covariances is convenient and simple if one considers that the process is $y = Cu$, with C being a matrix and u a vector of “components”. If we assume that they are independent, they will each have an autocovariance independent of the rest, and $E[u(x_1)u(x_2)^T]$ will be a diagonal matrix.

The last part of the video presents a Matlab example of said processes (2 outputs) and generates realizations of it.

7.1.4 Componentes principales (transformación autofunciones Karhunen-Loève)

[213: karhloevml] Análisis componentes principales de un proceso estocástico
15:52 (transf. Karhunen-Loève): ejemplo Matlab (muestreado)

Materiales: [[CÓD.:](#) pruebasKernelEigenvaluesV3.mlx] [[PDF](#)]

[[YouTube](#) ►]

Este vídeo considera 361 muestras de un proceso estocástico cuya densidad espectral de potencia viene dada por un filtro de Butterworth. La transformada inversa de Fourier produce la correlación $\kappa(x_1, x_2)$ estacionaria, y con ella se genera la matriz de varianzas-covarianzas de 361x361 de las muestras desde 0 a 36 segundos, con período 0.1 s.

Los autovectores y autovalores de la matriz sirven tanto para generar muestras de ese proceso entendido como una distribución normal 361-dimensional (se recomienda visualizar el vídeo [208] previamente a éste), como para hacer un análisis de componentes principales

(trayectorias que explican más varianza), haciendo un análogo de las ideas de análisis de series de datos empíricos (vídeos [710] y [711]) según su matriz de varianzas covarianzas, pero ahora aplicado a las series temporales.

Se comprueba que cuanto más lenta es la dinámica del proceso estocástico (menor ancho de banda del filtro), menos componentes son necesarios para explicar el 95% de la variabilidad. Las ideas están en cierto modo relacionadas con el teorema de Shannon sobre el muestreo (que usa la autocorrelación $\kappa(x_1, x_2) = \frac{\sin(x)}{x}$, y que aquí también se comprueban sus componentes principales).

Cuando el período de muestreo tiende a cero, los autovectores pasan a ser *autofunciones* del tiempo, y el análisis de componentes principales se convierte en lo que se denomina transformación de Karhunen-Loève en la literatura.

[214: gpkh2] Componentes Karhunen-Loeve (PCA) de proceso gaussiano, ejemplo Matlab
**** 15:19

Materiales: [[CÓD.:](#) GPpcaHKtestESP.zip] [[PDF](#)]
[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo presenta la obtención de los componentes principales (autofunciones de Karhunen-Loeve en el límite continuo) de un proceso estocástico gaussiano de kernel de covarianza ‘exponencial cuadrático’ (elegido arbitrariamente). Por simplicidad y para las representaciones gráficas se considerará un conjunto ‘discreto’ de 241 puntos de prueba donde analizar la estructura de la covarianza $K_{241 \times 241}$.

Los componentes principales, obtenidos a partir de la diagonalización $K = VDV^T$, son una de las muchas ‘raíces cuadradas matriciales’ $K = QQ^T$ que pueden calcularse. En concreto, son una raíz cuadrada ‘ortogonal’ en el sentido de que las columnas de $Q = V\sqrt{D}$ son ortogonales entre sí (componentes incorrelados).

El vídeo discute diversas cuestiones conceptuales relacionadas con raíces cuadradas matriciales y componentes principales, y finaliza representando gráficamente algunos de ellos.

El vídeo [215], continuación de este, hará una animación de la reconstrucción de una realización a partir de componentes principales ‘latentes’ normal standard, y también lo hará con un proceso del cual se hayan observado valores en ciertos puntos.

Otro ejemplo similar de análisis aparece en el vídeo [213].

Como la raíz cuadrada Q de la matriz de covarianza no es única, existen otras representaciones causal, anticausal y bilateral, que se analizan en los vídeos [??] y [??].

[215: gpkh2p] Componentes Karhunen-Loeve (PCA) de proceso gaussiano, animación y discusión adicional
**** 13:50

Materiales: [[CÓD.:](#) GPpcaHKtestESP.zip] [[PDF](#)]
[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo es continuación del vídeo [214], donde se planteaba el significado de una raíz cuadrada matricial de la covarianza $K = QQ^T$ y su límite continuo (autofunciones Karhunen-Loève). En este vídeo, se generan realizaciones añadiendo componente a componente multiplicados por una variable normal estándar, en una animación Matlab. La segunda parte del vídeo calcula el posterior suponiendo que se ha observado el proceso en algunos puntos (ver vídeo [210] para detalles). Luego, diagonalizando la matriz de covarianza, se analiza la forma de los componentes principales (autovectores) y se construyen realizaciones del posterior a partir de ellos.

Componente principales en procesos de varias salidas

[216: mimogp3pcaEN] (3/5) Multi-output Gaussian Processes: principal component analysis, Karhunen-Loeve eigenfunctions ****

09:46

Materiales: [[CÓD.:](#) GPmultioutputpart1.mlx] [[PDF](#)] [[YouTube](#) ►]

This video presents the PCA (principal component analysis) of the covariance matrix (at a finite set of points) of a 2-output gaussian process whose motivation and covariance structure details were discussed in previous videos [211] and [212]. When the number of points is infinity, in the continuous-input case principal components are Karhunen-Loeve (KL) eigenfunctions. See bottom note for references to one-output KL analysis.

In here, we exploit the special structure of 2-output Gaussian process to better understand such eigenfunctions via suitable plots, seeing that some eigenfunctions are equal in both outputs (strong common component) and only at higher frequencies eigenfunctions appear explaining the difference between the two outputs.

A simpler single-output Gaussian process PCA (KL) is studied in videos [??] and [??].

[217: gpcholEN] Gaussian process: Cholesky factor of covariance, spectral factor (Matlab example) ***** 19:34

Materiales: [[CÓD.:](#) GPcholENG.zip] [[PDF](#)] [[YouTube](#) ►]

This video presents the interpretation of the Cholesky factorization (lower triangular) of the covariance matrix of a Gaussian stochastic process (in discrete time, since we approximate a continuous process through a finite number of test abscissa points). An initial part reviews the basic ideas from the video [??], about the square roots of a covariance matrix and the principal components. Then, it focuses on the main objective, which is to interpret the meaning of the triangular structure of the Cholesky factor... it is given a 'causal' interpretation: the latent variable ' j ' has an effect on the outputs in the $j + 1, \dots, 241$ positions. Thus, a realization of the process 'from left to right' is constructed.

It is observed that the columns of Q converge to a single sequence, which moves downwards (when the steady state is reached and the finite window effects have disappeared). The video justifies why this gives rise to a *convolution* formula to calculate the effect of the latent variables on the outputs; The convolution kernel can be interpreted as the impulse response of a time-invariant linear system that is called 'causal spectral factor' in literature.

As square roots are not unique, there are other representations (anticausal, bilateral) of interest, which will be discussed in the video [218], a continuation of this one.

[218: gpanticaEN] Gaussian process: anticausal and bilateral representations (Matlab example) ***** 13:48

Materiales: [[CÓD.:](#) GPcholENG.zip] [[PDF](#)] [[YouTube](#) ►]

This video presents the 'anticausal' and 'bilateral' (mixture of causal and anticausal) representations of a Gaussian process, in a practical way via a Matlab example. The representations come from using a particular square root of the covariance matrix, since such matrix square root is not unique. The first three minutes review the orthogonal square root that gives rise to the Karhunen-Loeve principal components (video [??]) and the lower triangular Cholesky factorization that gives rise to the causal representation (video [217]).

By permuting rows and columns of the Cholesky decomposition, equivalent to a rearrangement of the random variables, we obtain an 'anti-causal' representation where each latent variable influences the output of the process in the same place and on the previous ones (past), following the way we decided to order the abscissas from the Gaussian process (which we could interpret as 'time'). This gives rise to an animation where a realization of the Gaussian process is built 'from right to left'.

The second part of the video discusses the symmetric square root $Q = V \cdot \sqrt{D} \cdot V^T$. In that case $K = QQ^T = Q^2$. The result is a 'mixture' of causal and anticausal (symmetric) action of the latent variables on the output, except effects of initial conditions or finite window; this will give rise to what would be a *bilateral convolution kernel*. An intuitive interpretation is that each row of Q (well, just the 'stationary' ones) is like a 'hammer blow' that deforms symmetrically on either side of the impact point, and that the intensity of the impact is the standard normally distributed latent variable.

7.2 Ecuaciones en diferencias estocásticas: Respuesta ante entrada aleatoria de sistemas discretos

Aunque las ecuaciones de la física son realmente en tiempo continuo, el análisis de los procesos estocásticos en tiempo continuo es un poco más difícil que el discreto. En sistemas en tiempo discreto, la simulación de procesos estocásticos con dinámica lineal se hace más comprensible y, por tanto, la presentación comienza por estos sistemas. Esta sección detalla, de forma "autocontenida" dicho caso sencillo. Intencionalmente, los vídeos siguientes pueden visionarse sin necesidad de haber visionado o comprendido en profundidad los vídeos de secciones anteriores de este capítulo.

El caso de tiempo continuo se abordará en la Sección sec:stoccont.

[219: stochOA]
Series Temporales y procesos estocásticos lineales (tiempo discreto)

 11:00

Materiales: [procestoc.pdf]
[YouTube ▶]

En este video se formaliza el concepto de "sistema lineal con perturbaciones aleatorias" en términos de "proceso estocástico lineal", en tiempo discreto. Se discute:

- Motivación: series temporales.
- Ruido blanco versus ruido coloreado.
- Procesos en representación interna: ruido de proceso y ruido de medida.
- Simulación de la ecuación de estado: ecuación de medias y ecuación de varianzas.
- Ecuación de salida en medias y varianzas
- Ejemplo numérico: paseo aleatorio $x_{k+1} = x_k + \nu_k$, con $\nu_k \sim \mathcal{N}(0, 1)$. Intervalos de confianza en el tiempo. Este proceso es muy importante en la generalización de las ideas aquí presentadas (ecuaciones en diferencias estocásticas) a tiempo continuo (ecuaciones diferenciales estocásticas). La versión en tiempo continuo se denomina "Proceso de Wiener", límite de este integrador cuando el período de muestreo tiende a cero, ver vídeo [223].

7.2.1 Ejemplos

[220: tuborui]
Análisis de procesos lineales sujetos a entradas tipo ruido coloreado: ejemplo Matlab tubería

 12:29

Materiales: [CÓD.: ModeloTuboTermicoEstocas.mlx] [PDF]
[YouTube ▶]

En este vídeo se analiza el efecto sobre un proceso lineal de entradas tipo “ruido coloreado”. El proceso es el discutido en el vídeo [273]: una tubería por la que pasa gas que la calienta, y una sonda sobre dicha tubería. En ese vídeo se analizó el detalle del modelado, y la respuesta ante escalón y en frecuencia.

El objetivo ahora es simular una entrada de temperatura T_{gas} que sea un ruido “coloreado”. Este ruido se modelará como un ruido blanco que pasará por un filtro $1/(s + \alpha)^2$, que al ser un filtro paso-bajo (vídeo [195]), aproximadamente sólo mantendrá los componentes frecuenciales por debajo de α en la densidad espectral de potencia de T_{gas} ; el concepto se define en el vídeo [207]. Cuanto más pequeño sea el ancho de banda α , más “lenta” será la variación entre unos valores y otros de la temperatura del gas. Nótese que al pasar por ese filtro, T_{gas} dejará de ser una entrada y pasará a ser una salida del nuevo sistema (la entrada ya será un ruido blanco –artificial–).

El proceso se discretiza a 0.2 s de período de muestreo, cien veces por encima de $\alpha = 0.16$ por lo que la selección del período (Tma. Shannon, vídeo [369]) parece acertada y suponer constante entre muestreos (retenedor orden cero) también (hay otras formas teóricamente más correctas de discretizar ruidos continuos, con las fórmulas exponenciales de ecuaciones de varianzas discutidas en el vídeo [233]).

Al modelo discreto, se le calcula la matriz de varianzas-covarianzas estacionaria del estado con el comando `gram`, se usa para calcular las varianzas de las temperaturas de sonda, tubo y gas (salidas), y las desviaciones típicas que son interpretables (en °C). El comando `norm(·, 2)` también puede calcular dicha desviación típica (norma 2 de un sistema, vídeo [633]).

En la parte final, se realizan simulaciones temporales de realizaciones (entrada `randn`), y se comparan con los intervalos de confianza obtenidos por la solución de varianza estacionaria (y también la no estacionaria, vídeo [219]). También se simula el filtro $1/(s + \alpha)$ de primer orden para generar T_{gas} y se observa que, obviamente, tiene más componentes de alta frecuencia.

Si se dispone de un sensor de la temperatura de la sonda, el filtro de Kalman (estacionario) asociado se discute en el vídeo [605].

[221: stochML]

Simulación de procesos estocásticos lineales (discretos): ejemplo

Matlab masas-muelles-amortiguadores con ruido

Materiales:

C 17:53

[YouTube ►]

[CÓD.: [simustochmasamueller.mlx](#)]

[PDF]

En este vídeo se presenta la simulación de un proceso lineal (dos masas, muelles, amortiguadores) de orden 4. Por simplicidad, se simula en tiempo discreto. Un ejemplo de modelado más complejo obteniendo un modelo como el que aquí se simula, a partir de un sistema con muelles no lineales, aparece en el vídeo [88].

El componente estocástico se añade suponiendo que se ejercen fuerzas sobre las masas aleatorias con una cierta desviación típica y distribución normal, discretizando con ‘zoh’... realmente si las fuerzas son un ruido blanco en tiempo continuo, esta discretización no es del todo correcta (ver por qué no lo es en el vídeo [222]), y deberían usarse fórmulas exponenciales discutidas en el vídeo [233]: la varianza del ruido de proceso debería depender del período de muestreo al que se haya discretizado. No obstante, esto rebasa los objetivos “simples” de entender procesos discretos y, aunque no sea formalmente 100% correcto (el tiempo *continuo* es mucho más complejo que el discreto cuando hay ruido), lo que se hace aquí es una aproximación más o menos simplista pero razonable.

El vídeo compara la simulación en media y varianzas (a partir de las cuales se obtienen intervalos de confianza de las salidas) con la realización de una simulación dando valores concretos (`randn()`) a las perturbaciones, para comprobar las predicciones de la teoría. También se analizan las formas que provee el Control System Toolbox de Matlab de calcular la varianza estacionaria en procesos estables, con los comandos `dlyap` (ecuación

de Lyapunov discreta) y **gram** (gramiano de controlabilidad, ver vídeo [279] para ampliar dicho concepto).


7.3 Ecuaciones diferenciales estocásticas (procesos estoc. en tiempo continuo)

Motivación:

[222: edostochmal]

11:14

Procesos continuos con ruido (procesos estocásticos en tiempo continuo): como NO simularlos bien (+truco para hacerlo correctamente)

*** 

Materiales: [[CÓD.: estocODEcomoNOhacerlo.mlx](#)] [[PDF](#)]

[YouTube ▶]

Ante un modelo de un proceso continuo (por ejemplo $1/s$ en este vídeo), la primera opción para simularlo sería, por ejemplo, usar `randn()` como entrada al período de muestreo más pequeño posible.


En este vídeo, un ejemplo de Matlab comprueba que cuando el período de muestreo tiende a cero, entonces, la solución simulada tiende a cero. Necesita una corrección (dividir por la raíz cuadrada del período de muestreo o paso de integración) que aquí se plantea como una idea *ad hoc*, pero que será justificada adecuadamente en otros materiales.

En concreto, el integrador aquí estudiado se denomina *Proceso de Wiener*, y la teoría subyacente se discute en el vídeo [223], y otro ejemplo Matlab aparece en el vídeo [224]. Una vez introducidas esas ideas, la justificación teórica final de cómo simular correctamente las *ecuaciones diferenciales estocásticas* con el truco de la raíz cuadrada del paso de integración se presenta en el vídeo [226]. Este método de simular procesos estocásticos se denomina Euler-Maruyama en la literatura.

7.3.1 El proceso de Wiener (movimiento Browniano)

[223: wienerlim]

Procesos Estocásticos, proceso de Wiener: límite tiempo continuo de integrador en tiempo discreto

*****  18:52

Materiales: [WienerFromDiscrete.pdf]

[YouTube ▶]

En este vídeo se plantea describir un proceso continuo como límite de un integrador en tiempo discreto $x_{k+1} = x_k + w_k$, cuando se acumula ruido en períodos infinitesimales de tiempo. Ese proceso es el “proceso de Wiener” o “movimiento browniano” unidimensional.

Se analizar, basado en la fórmula de la varianza de la suma de variables aleatorias por qué la varianza del incremento de x debe ser igual al período de muestreo (desviación típica raíz cuadrada del período de muestreo), de modo que el proceso de Wiener verifica que, tendiendo el tiempo a cero (incrementos infinitesimales), sus incrementos dx son variables aleatorias $dx \sim N(0, dt)$, distribución normal de media cero y varianza dt , desviación típica \sqrt{dt} .

Ejemplos Matlab de las ideas desarrolladas aquí se ilustran en el vídeo [224], continuación de éste. Un análisis en detalle de diversas propiedades de este proceso y de la interpretación física se aborda en el vídeo [225].

Si en vez de un *integrador* puro se aplica la misma idea de límite a un proceso con otra dinámica, eso da lugar a la definición de las ecuaciones diferenciales estocásticas $dx = f(x)dt + g(x)dB$, con $dB \sim N(0, dt)$, analizadas en el vídeo [226].

[224: wienermlab1] Procesos Estocásticos, proceso de Wiener: ejemplo Matlab 08:32

*** **Materiales:** [[Cód.:](#) browniano1dA.m] [[PDF](#)]

[YouTube ▶]

Este vídeo presenta dos formas de visualizar el proceso de Wiener (movimiento browniano) unidimensional, basándose en las consideraciones teóricas del vídeo [223]:

- Como un proceso “continuo” subyacente que se ve cada vez con más frecuencia de muestreo, viendo que los incrementos tienen desviación típica $\sqrt{T_s}$.
- Como un integrador discreto $x_{k+1} = x_k + w_k$ que se ejecuta 1, 2, ..., N veces en el intervalo $[0, 1]$.

[225: wienerprops] Procesos Estocásticos, proceso de Wiener: Interpretación física y propiedades 19:50

***** **Materiales:** [WienerPropiedades.pdf]

[YouTube ▶]

Este vídeo discute la interpretación física y las propiedades del proceso de Wiener, descrito en más detalle en los vídeos previos [223] y [224], cuya visualización se aconseja antes de visualizar el presente material.

Las propiedades analizadas son:

- Media, Varianza y Desviación típica del proceso y sus incrementos.
- Covarianza entre distintos instantes
- Auto-similaridad (al estilo de los *fractales*)
- Continuidad (con probabilidad 1)
- Diferenciabilidad (más bien la ausencia de ella)

La interpretación física hace el nexo entre los fenómenos “microscópicos” de movimiento browniano de partículas individuales con los fenómenos “macroscópicos” de difusión o transmisión de calor. De hecho existe una formulación alternativa de la dinámica de procesos estocásticos con ecuaciones en derivadas parciales para calcular la derivada temporal de la función de densidad. En el proceso de Wiener, coinciden con ecuaciones de difusión bien conocidas.

Los incrementos del proceso de Wiener serán la entrada a modelos de ecuaciones diferenciales estocásticas más generales que el integrador puro, como se discute en el vídeo [226].

Alternativamente, el proceso de Wiener es la EDO estocástica más sencilla (integrador puro), caso particular de modelos lineales multivariables definidos en el vídeo [227].

El “doble integrador” (integrated Wiener process) y sus propiedades se discuten en el vídeo [237].

7.3.2 Ecuaciones Diferenciales Estocásticas

[226: edostoch] Procesos estocásticos en tiempo continuo: representación como ecuaciones diferenciales estocásticas

*****  16:18**Materiales:** [EDOestocascontinuo.pdf]

[YouTube ▶]

Este vídeo generaliza el proceso de Wiener $dB \sim N(0, dt)$ (incremento infinitesimal con desviación típica \sqrt{dt} , descrito en los vídeos [223] y [225]), a un proceso general $dx = f(x, t)dt + g(x, t)dB$, denominado ecuación diferencial estocástica. La simulación de estas EDO estocásticas genera realizaciones $x(t)$ de un proceso estocástico en tiempo continuo X_t .

Esta expresión es la representación formalmente correcta (en el sentido de Itô) de la idea de *ecuación diferencial con ruido* $\frac{dx}{dt} = f(x, t) + g(x, t)w$, siendo w un “ruido blanco” que tiene varianza infinita.

El vídeo discute los detalles de cómo interpretar y simular (integración numérica) este tipo de ecuaciones diferenciales estocásticas. La simulación a un cierto paso de integración T puede ser bien en un script de Matlab, generando el `randn()` multiplicado por las desviaciones típicas adecuadas, bien en Simulink mediante un bloque **Band-Limited White Noise**, bien usando integradores numéricos usuales (`ode45`, `lsim`, ...). Los detalles de todo eso son explicados en este material.

En resumen, se presenta la justificación a la idea que se lanzó de modo “a ojo” en el vídeo [222] sobre determinados escalados de `randn()` dependientes de la raíz cuadrada del paso de integración.

Este método de simular procesos estocásticos se denomina Euler-Maruyama. Igual que con las ecuaciones diferenciales ordinarias, hay otros métodos más exactos para poder aumentar el período de muestreo conservando precisión... pero no son objetivo de este vídeo ni de, por el momento, ninguno de la colección, dado que nos centramos en casos lineales cuya solución exacta puede ser calculada con exponenciales.

En efecto, el caso particular *lineal* se discute en el vídeo [227], por su importancia en ingeniería de control y de comunicaciones.

7.3.3 EDO Estocásticas lineales

[227: edoslin1] **Ecuaciones diferenciales estocásticas lineales**
(Ornstein-Uhlenbeck): definición y conceptos preliminares

Materiales: [EDOestocascontinuoLineal.pdf]

*** 13:12

[YouTube ▶]

Este vídeo presenta las ecuaciones diferenciales estocásticas lineales $dx = (Ax + Bu)dt + GdW$ como un caso particular de la ecuación diferencial estocástica general $dx = f(x, t)dt + g(x, t)dW$ cuyo significado y simulación se detallaban en el vídeo [226], que generan series temporales en tiempo continuo.

Aquí se plantea el caso multivariable, con $E[dWdW^T] = I \cdot dt$ o con posible correlación $E[dWdW^T] = \mathcal{W} \cdot dt$, siendo \mathcal{W} una matriz, discutiendo las dimensiones de las diferentes matrices y cómo simular en Matlab este tipo de sistemas; realmente, no obstante, no hay ningún concepto fundamentalmente diferente de los ya discutidos en el vídeo referido en el párrafo anterior.

Este vídeo presenta las ideas preliminares que justifican la importancia de la simulación de la evolución temporal de la media y la varianza, que serán objeto de desarrollo específico en los vídeos [228] y [229], respectivamente.

[228: edoslinm] **Ecuaciones diferenciales estocásticas lineales: ecuación de medias**

Materiales: [EDOestocascontinuoLineal.pdf]


**** 07:11


[YouTube ▶]

Este vídeo es continuación del vídeo [227], resumido aquí en los primeros dos minutos, sobre cálculo estocástico en EDO estocásticas lineales.

Aquí se desarrolla lo que se conoce como *ecuación de medias*, que, según se argumenta, acaba siendo la simulación de un proceso lineal suponiendo ruido igual a cero.

El desarrollo continua con las ecuaciones de varianzas, discutidas en el vídeo [229].

[229: edoslinv] Ecuaciones diferenciales estocásticas lineales: ecuación de varianzas en tiempo continuo *****  22:14

Materiales: [EDOestocascontinuoLineal.pdf] 

Este vídeo es continuación de los vídeos [227] y [228], resumidos aquí en los primeros dos minutos y medio, sobre cálculo estocástico en EDO estocásticas lineales $dx = (Ax + Bu)dt + GdW$, $E[dWdW^T] = \mathcal{W} \cdot dt$.

Aquí se desarrolla lo que se conoce como *ecuación de varianzas*, una ecuación que describe la evolución temporal de la matriz de varianzas-covarianzas de un proceso multivariable lineal sujeto a entradas de ruido, siendo el resultado $\dot{P} = AP + PA^T + GWG^T$.

La parte final del vídeo discute cómo evoluciona la autocovarianza entre dos instantes de tiempo de una realización, $x(t)$ y $x(t + \tau)$.

El vídeo [230] discute cómo integrar con Matlab y `ode45` las ecuaciones de medias y varianzas discutidas aquí. La solución exacta con fórmulas exponenciales se aborda teóricamente en el vídeo [233], y en otros ejemplos Matlab que siguen a dicho vídeo.

El caso discreto de ecuaciones de medias y varianzas es bastante más sencillo conceptualmente (vídeo [219]).

Ejemplos Matlab

[230: edoslim1] Ecuaciones diferenciales estocásticas lineales: integración ode45 de ecuaciones de media y varianzas ***  15:21

Materiales: [[CÓD.:](#) SimulaEDOSTochLineal.mlx] [[PDF](#)] 

En este vídeo se describe el código Matlab que realiza la integración numérica directa de las ecuaciones de medias y varianzas asociadas a una EDO estocástica lineal, según la teoría vista en el vídeo [228] (medias) y el [229] (varianzas).

Realmente, también existen soluciones exactas con exponenciales de matrices, pero aquí se persigue entender el significado de ellas de forma directa, integrando con `ode45` y graficando el resultado.

El vídeo [231], continuación de este, dibujará los intervalos y elipsoides de confianza asociados a las soluciones obtenidas aquí.

[231: edoslim2] Ecuaciones diferenciales estocásticas lineales: intervalos y elipsoides de confianza via integración ecs. media y varianza *****  13:52

Materiales: [[CÓD.:](#) SimulaEDOSTochLineal.mlx] [[PDF](#)] 

Este vídeo es continuación del [230], donde se detallaba el código para integrar las ecuaciones de medias y varianzas de una EDO estocástica lineal (teoría en vídeos [228] y [229]) que modela un sistema masa-muelle-amortiguador (muy poco amortiguado, para que la autocorrelación dure mucho, intencionalmente) sujeto a aceleración ruido blanco.


Aquí se plantea dibujar la media \pm un múltiplo de la desviación típica, para tener una idea de un intervalo de confianza.

También se deforma un círculo (región de confianza) obtenido con las fórmulas χ^2 para la suma de dos distribuciones normales estándar, multiplicándolo por la raíz cuadrada (matricial, `sqrtn`) de la matriz de varianzas-covarianzas resultado de la integración numérica. Con ello, se dibuja la evolución en el tiempo de la región de confianza en el *plano de fase* (posición-velocidad). La evolución del elipsoide de confianza en distintos escenarios de inicialización de la varianza se ilustra en el vídeo [232], cuyo código es muy similar al aquí usado para dibujar los elipsoides.

La última parte del vídeo se dedica a simular realizaciones individuales, tirando un dado normal con desviación típica \sqrt{T} para un paso de integración T suficientemente pequeño, según discutido en el vídeo [226]. Si se simulan 600 realizaciones, se comprueban (contando) los intervalos de confianza calculados teóricamente con las ecuaciones de medias y varianzas (obviamente, en esas simulaciones de medias y varianzas NO se tira ningún dado).

[232: solectvzml] [Simulación ode45 ec. varianza en proceso continuo segundo orden: ejemplo Matlab elipsoide confianza](#)

Materiales: [[CÓD.: animasimulaEDOStochLineal2.m](#)] [[PDF](#)]

****  11:33

[YouTube ▶]

Este vídeo ilustra la evolución en el tiempo de los elipsoides de confianza asociados a la ecuación de varianzas de un proceso estocástico en tiempo continuo, integrando numéricamente (ode45) la ecuación de varianzas. Este vídeo complementa los conceptos ilustrados en el vídeo [231] cuya visualización se aconseja.

Aquí se presenta el elipsoide de confianza empezando:


- desde "cero" (proceso en reposo), con lo que el elipsoide aumenta de tamaño,
- desde una región inicial "muy grande" (entonces el elipsoide de confianza se contrae) y
- un tercer caso donde se supone conocida con precisión una de las variables de estado y entonces se observa un elipsoide que sólo se hace más ancho en una dirección conforme pasa el tiempo, y que "gira" de modo que la incertidumbre máxima/mínima se va trasladando periódicamente de la variable x_1 a la x_2 y viceversa. Esta última idea se explora en más profundidad en el vídeo [239].

Aparte de la integración "numérica" de las varianzas, existen fórmulas exponenciales para la obtención exacta (en el caso lineal) de la varianza en función del tiempo; la teoría implicaría compilar a numérico expresiones con integrales de exponenciales de matrices (vídeo [233]), pero existen expresiones equivalentes donde sólo hace falta calcular numéricamente la exponencial de una matriz (vídeo [234]).

7.3.4 Solución exponencial de ecuaciones de medias, varianzas y autocovarianzas

[233: solectmv] [Ecuaciones diferenciales estocásticas lineales: solución ecuaciones medias y varianzas \(expm\)](#)

Materiales: [SolEcMedVzas.pdf]

****  13:13

[YouTube ▶]

Este vídeo presenta la solución exacta de las ecuaciones diferenciales de medias y varianzas asociadas a un proceso estocástico lineal sujeto a entradas de ruido. Se utiliza la técnica del "factor integrante" multiplicando por las adecuadas exponenciales de matrices para obtener la solución a las citadas ecuaciones diferenciales ordinarias.

La parte final del vídeo también detalla la exponencial de matriz necesaria para el cálculo de la auto-covarianza entre muestras del proceso en diferentes instantes de tiempo.

Ejemplos de aplicación aparecen, por ejemplo, en los vídeos [235], [237], [238] y otros.

En código Matlab no es necesario hacer, en muchos casos, las integrales de exponenciales que resultan de las expresiones teóricas aquí; con código de exponencial de matriz puede ser suficiente, en lo que se denomina fórmula de Van Loan, cuya obtención se detalla en el vídeo [234].

[234: vanloan] Fórmula de Van Loan (expm) para discretización de ecuación de varianzas de ODE estocástica lineal **** 13:55
Materiales: [DiscretizRuidoVanLoan.pdf] [YouTube ▶]

*LINK TO ENGLISH VERSION

Este vídeo discute la fórmula de Van Loan (<https://doi.org/10.1109/TAC.1978.1101743>) para calcular mediante una única exponencial de matriz la integral de exponenciales de matrices que es la solución de la ecuación de varianzas de un proceso estocástico lineal sometido a ruido de densidad de potencia constante en el tiempo (teoría en vídeos [229] –planteamiento de la ecuación– y [233] –solución exponencial–).

Esta es la fórmula usada en los comandos `kalmd` y `lqrd` de Matlab.

Ejemplos Matlab, simulación y predicción

[235: edoslin1oa] Ecuaciones de medias y varianzas en procesos de primer orden: *** 12:45
 ejemplo Matlab
Materiales: [Cód.: EjemplosEcVarianzasORDEN1.mlx] [PDF] [YouTube ▶]

Este vídeo discute la ecuación de varianzas en sistemas de primer orden, analizando los resultados de la solución exponencial en el sistema $dx = Adt + dW$, con $A = -1$ (estable), $A = +1$ (inestable), y $A = 0$ (integrador puro, o sea el proceso de Wiener). Los detalles teóricos están en el vídeo [233].

Se representa la desviación típica en función del tiempo (asumiendo cond. inicial cero), y se discute el régimen estacionario de la varianza (que sólo existe en el caso estable).

El vídeo [236] incorpora la autocovarianza entre distintos instantes de tiempo, para hacer predicciones.


[236: edoslin1ob] Ecuaciones de covarianza en procesos de primer orden: **** 16:13
 interpolación y extrapolación (Matlab)
Materiales: [Cód.: EjemplosEcVarianzasORDEN1.mlx] [PDF] [YouTube ▶]

Este vídeo resume brevemente el vídeo [235] y, sobre el mismo ejemplo $dx = Adt + dW$, con $A = -1$ (estable), $A = +1$ (inestable), y $A = 0$ (integrador puro, el proceso de Wiener) calcula la covarianza $S(t, \tau)$ entre $x(t + \tau)$ y $x(t)$, con $\tau \geq 0$.

Esa fórmula de covarianza (así como la de la varianza) es utilizada para resolver dos problemas de interés:

- Extrapolación: predecir *a futuro* la evolución más probable (e intervalo de confianza) dada una medida en un cierto instante;
- Interpolación: predecir *a toro pasado* cuál es la evolución pasada más probable que ha dado lugar a una cierta medida (suponiendo conocido que $x(0) = 0$; en el caso estable esta suposición se puede ignorar esperando el suficiente tiempo a que alcance condiciones estacionarias antes de efectuar la medida).

Las ideas de interpolación y extrapolación basadas en modelos estadísticos dan lugar a lo que se conoce como “estimación en procesos gaussianos”, y diferentes filtros e interpoladores. Es una teoría muy extensa con múltiples aplicaciones. En esta colección se aborda, por ejemplo, en el vídeo [240], [241] y otros.

[237: iwp1] Posición sujeta a aceleración aleatoria (integrated Wiener Process) propiedades y predicción estadística sobre el mismo (Matlab) ****  19:24

Materiales: [[CÓD.:](#) IntegratedWienerProcess.mlx] [[PDF](#)] [[YouTube](#) ►]


Este vídeo discute las conclusiones que se pueden obtener a partir de las ecuaciones de medias y varianzas (y autocovarianza) de la integral del proceso de Wiener, que físicamente modela a una partícula sometida a aceleración aleatoria (sin pérdidas por rozamiento, doble integrador).

Se observa que la varianza de la posición es $t^3/3$, que hay correlación entre posición y velocidad en el mismo instante, y que la ecuación de medias se reduce a las ecuaciones de un movimiento uniforme (aceleración cero).

Con las fórmulas de varianzas y covarianzas en el mismo y en diferentes instantes de tiempo, se predice velocidad dada posición, y también se abordan los problemas de “extrapolación” e “interpolación” dada una medida, análogos a los abordados en el vídeo [236] para procesos de primer orden, pero esta vez con un proceso de orden 2. Se aconseja visualizar previamente el referido vídeo dado que el caso de primer orden es más sencillo de entender.

La parte final del vídeo aborda la relación del “integrated Wiener process” aquí estudiado con las “splines cúbicas” (polinomios cúbicos de interpolación).

La generalización de las ideas lanzadas aquí a múltiples medidas se discute en el vídeo [242], continuación de éste.

[238: dto2] Solución ec. varianzas: fórmulas exponenciales y gramiano (ejemplo Matlab) ***  13:07

Materiales: [[CÓD.:](#) MedidaPosicionEstimarLuegoVelocidad.mlx] [[PDF](#)] [[YouTube](#) ►]

Este vídeo aplica las fórmulas exponenciales desarrolladas en la teoría (vídeo [233]) a un sistema de orden 2, que modela un masa-muelle-amortiguador con aceleración tipo ruido blanco.


Se discute el concepto de gramiano, varianzas-covarianzas en régimen estacionario y se aborda la integración de ecuaciones de medias y varianzas durante un período T_s , de modo que con ello se tendrán las ecuaciones de medias y varianzas de un proceso discreto equivalente (cada T_s segundos).

Se comparan las exponenciales con el desarrollo de Taylor de orden 1 en el período de integración, observándose que coinciden cuando T es pequeño, y que equivaldrían a integrar por Euler las ecuaciones diferenciales en tiempo continuo de medias y varianzas, como se ha hecho en el vídeo (230), bueno en ese vídeo con `ode45`.

Si el paso de integración (período de discretización) no es pequeño, entonces la solución exacta exponencial ya es diferente de la integración numérica por Euler.

La raíz cuadrada matricial (`sqrtm`) de la integración del efecto del ruido $\int_0^{T_s} e^{A\tau} B_w B_w^T e^{A^T \tau} d\tau$ sería necesaria para simular realizaciones concretas. Se observa que aunque en continuo el ruido sea de aceleración, la simulación discreta requiere dos `randn` para ser simulado de forma correcta.

Este vídeo continua en el [239], donde estas discretizaciones son utilizadas para simular intervalos de confianza y realizaciones concretas.

[239: mpvr] Simulación (exacta) de ecs. de medias y varianzas en proceso continuo segundo orden: ejemplo Matlab ****  20:59

Materiales: [[CÓD.:](#) MedidaPosicionEstimarLuegoVelocidad.mlx] [[PDF](#)] [[YouTube](#) ►]


Este vídeo utiliza fórmulas exponenciales para discretizar un proceso continuo de modo que se pueda hacer una simulación exacta en los instantes múltiplos del período de muestreo. La teoría se ha visto en el vídeo [228] (medias) y [229], y una discusión más detallada sobre el significado de esas exponenciales y su comparación con la integración numérica (para el mismo modelo de proceso que el utilizado aquí) se aborda en el vídeo [238], cuya visualización podría ser aconsejable.

En este vídeo se revisan los contenidos del vídeo [238] en los tres primeros minutos, y luego se representan gráficamente los resultados de simular la discretización, en términos de media e intervalo de confianza. Obviamente, se podrían representar elipsoides de confianza también con la discretización exacta, al igual que el vídeo [232] lo ha hecho con integración `ode45`. Quizás visualizar este vídeo previamente al que aquí estamos describiendo sea también aconsejable.

Aquí se supone que se dispone de una medida exacta de la posición para inicializar las simulaciones de la ecuación de varianzas (idéntico al caso 3 del vídeo [232]).

Se observa que en el proceso estudiado, cuya frecuencia natural tiene un período de 8 segundos, cuando ha transcurrido un cuarto de período entonces se produce la menor anchura del intervalo de confianza de la velocidad. La conclusión es que si se dispusiera de una única medida de posición, pero se deseara estimar velocidad, entonces se debería medir la posición 2 segundos antes del instante en el que se desea tener un estimado lo más preciso posible de la velocidad; esto es debido al “giro” del elipsoide de varianza por polos complejos, como se veía en el ya referido caso 3 del vídeo [232].

7.4 Estimación en series temporales a partir de Kernel de covarianza

[240: krigt] **Kriging: estimación (interpolación/extrapolación) en series temporales con autocovarianza conocida** ****  08:44
Materiales: [GaussianStatisticInterp.pdf] [YouTube ►]


En este vídeo se plantea cómo, a partir de la función de autocovarianza $\kappa(t_1, t_2)$ de un proceso estocástico temporal, construir la mejor predicción de $y(t)$ para t arbitrario a partir de muestras $\{y_1, y_2, \dots, y_N\}$ tomadas en instantes $T = \{t_1, t_2, \dots, t_N\}$. Se utiliza la fórmula de predicción lineal óptima (vídeo [967]).

De hecho, la idea es una generalización del código preliminar en el vídeo [236] que puede servir de ejemplo Matlab introductorio a lo que aquí se plantea. Otro ejemplo se desarrolla en el vídeo [241], continuación de éste.

En el caso gaussiano, la fórmula de la varianza del error de predicción permitiría obtener intervalos de confianza a partir de las fórmulas de la distribución normal.

Se aborda el caso estacionario y su interpretación frecuencial posibilitada por las ideas de densidad espectral de potencia (vídeo [207]), y la incorporación de posible ruido de medida en las muestras no correlado con $y(t)$ ni con los otros instantes.

Nota: el vídeo [607] estudia un problema similar con modelos en representación interna.

[241: krigtm] **Interpolación/extrapolación en series temporales estacionarias a partir de número finito de muestras: ejemplo Matlab (Kriging)** ***  10:29
Materiales: [Cód.: GaussProcFiniteSamplesFilter.mlx] [PDF] [YouTube ►]

Este vídeo presenta un ejemplo Matlab de la metodología de interpolación/extrapolación de una señal generada por un proceso estocástico estacionario de autocovarianza (o densidad espectral de potencia, en caso estacionario, vídeo [207]) dada, cuya teoría se discutió en el anterior vídeo [240].

En concreto, para un cierto conjunto de muestras tomadas a intervalos irregulares (refinamientos de estas ideas pueden utilizarse en lo que se denomina control de sistemas con *muestreo no convencional*) se ilustran los resultados en media y en intervalo de confianza (proceso gaussiano) suponiendo que la dinámica que ha generado la señal ha sido un ruido blanco (densidad de potencia 1 a todas las frecuencias) excitando a un proceso lineal de orden 1, a un filtro Butterworth de segundo orden y a un filtro resonante de segundo orden. Si se trabajara con funciones de transferencia discretas $G(z)$ y transformada discreta de Fourier podrían hacerse desarrollos similares en tiempo discreto, que se dejan al lector (obviamente, las muestras tendrían que estar tomadas en múltiplos del período básico de muestreo para el que se haya descrito la función de transferencia).

Un caso de estimación en series temporales suponiendo que el proceso estocástico subyacente es no estacionario se aborda en el vídeo [242].

[242: wienkrig]

Estimación (interpolación, kriging) con procesos no estacionarios: Wiener, Integrated Wiener (ejemplo Matlab)

Materiales: [[CÓD.:](#) WienerRegression.mlx] [[PDF](#)]

**** 10:59

[YouTube ►]

Este vídeo plantea otros problemas de estimación (interpolación y extrapolación, kriging) en procesos gaussianos unidimensionales (esto es, podrían ser interpretados como series temporales).

Primero, se plantean dos kernels ($e^{-|d|}$ y e^{-d^2}), $d = x_2 - x_1$ estacionarios, para revisar conceptos de generación de muestras (detalles en vídeo [208]) y de estimación cuyo detalle se aborda en, por ejemplo, el vídeo [240] y más ejemplos Matlab aparecen en el vídeo [241].

En la segunda mitad del vídeo, se plantea la estimación con muestras de dos procesos estocásticos no estacionarios muy habituales:

1. El proceso de Wiener (paseo aleatorio, vídeo [223]), informalmente descrito como $\frac{dw}{dt} = \beta$, siendo β un ruido blanco de densidad de potencia unidad,
2. La integral del proceso de Wiener anterior, informalmente $\frac{d^2x}{dt^2} = \beta$, como una masa de 1 Kg con aceleración ruido blanco (el proceso de Wiener sería la velocidad de dicha masa; estamos suponiendo modelos integrador puro, sin fricción). Se aconseja visualizar el vídeo [237] para comprender en más detalle el proceso de Wiener integrado al que aquí se referencia.

Las fórmulas son las mismas aunque la matriz de VC no corresponda a un caso estacionario (matriz Toeplitz, todo trasladado alrededor de la diagonal). Se comprueba que:

1. El resultado de la estimación suponiendo muestras generadas por proceso de Wiener es una interpolación lineal y extrapolación manteniendo constante la última muestra;
2. El resultado de la estimación suponiendo muestras generadas por la integral del proceso de Wiener tiene derivadas primera y segunda continuas, y la derivada tercera es constante a trozos: dicha estimación es, por tanto, representada por un polinomio de grado 3 a trozos, que se denomina *cubic spline* en la literatura: el “*integrated Wiener process*” justifica en términos probabilísticos el uso de ciertas splines cúbicas para interpolación.

Nota 1: El caso no estacionario puede resolverse de forma equivalente si el proceso estocástico es modelado en representación interna y se aplica el algoritmo de suavizado Rauch-Tung-Striebel en tiempo continuo. Por simplicidad, en el vídeo [607] se explica el caso discreto, con ejemplos también discretos en los vídeos [613] y [632] que usan el modelo del doble integrador (como implícitamente se está haciendo aquí).

Nota 2: En los vídeos [746] y [747] se utilizan los procesos de Wiener y Wiener integrado aquí discutidos con una transformación exponencial de los datos de entrada para un problema de identificación de respuesta impulsional. Las propiedades del proceso de Wiener integrado (doble integrador) se discuten en el vídeo [237].

7.4.1 Estimación con medidas de “velocidad” (derivadas parciales de proceso estocástico)

[243: gradgp] Media y covarianza de las derivadas parciales (velocidad, gradiente) de un proceso estocástico

Materiales: [estimgradientofGPTheorySPAIN.pdf]

***** 15:55

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Ciertos procesos estocásticos (de “segundo orden” o superior, sea lo que sea eso, que en este momento no abordaremos) son diferenciables... esa derivada es, por ejemplo, la velocidad de una masa que se mueve sujeta a aceleraciones aleatorias; esa velocidad es un proceso estocástico. La posición es la integral de la velocidad, pero la velocidad es la derivada de la posición.

Definir formalmente la derivada de un proceso estocástico requiere cierto rigor (convergencia en probabilidad o en media cuadrática) pero, bueno, si hacemos “acto de fe” y creemos que entendemos lo que es la velocidad arriba mencionada, entonces a partir de las características estadísticas de la posición en media y varianza (función de covarianza) se pueden calcular las características de media y varianza, y covarianza velocidad-posición.

En un proceso estocástico $f(x)$ multivariable (no serie temporal $f(t)$ como antes parecía asumirse implícitamente), la idea se generaliza a estimar derivadas parciales de dicho proceso $f: \mathbb{R}^n \rightarrow \mathbb{R}$ (si existen), en media y varianza, a partir de $k(x, x')$.

Este vídeo comprueba que el gradiente de la covarianza $k(x, x')$ es la covarianza entre el proceso y su derivada, y que el hessiano (segundas derivadas) de la covarianza nos proporciona la covarianza entre las derivadas parciales en diferentes puntos.

La aplicación de estas fórmulas al caso de procesos estacionarios $k(x, x') = \kappa(x' - x)$ se discute en el vídeo [244].

Por supuesto, en series temporales también se podrían hacer operaciones con ecuaciones diferenciales estocásticas asociadas (por ejemplo, filtro de Kalman) o con la densidad espectral de potencia (caso estacionario).

[244: gradgpst] Caracterización derivadas de proceso estocástico con jacobiano/hesiano covarianza: caso estacionario

Materiales: [estimgradientofGPStationarySPAIN.pdf]

***** 11:10

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo particulariza las fórmulas del vídeo [243] sobre (primeras) derivadas de un proceso estocástico al caso estacionario, donde la covarianza $k(x, x')$ sólo depende de la diferencia entre los puntos $k(x, x') = \kappa(x' - x)$. Realmente es una aplicación directa de dichas fórmulas, pero dada la frecuencia con que el caso estacionario aparece en aplicaciones, es aconsejable deducirlas explícitamente.

**Errata:* cuando en el audio se dice derivada i -ésima o j -ésima se refiere a la “derivada primera en la dirección i ” o en la dirección j ... sólo se consideran primeras derivadas en este vídeo. Evidentemente, segundas derivadas y sucesivas requerirían aplicar las fórmulas sobre los resultados de la primera derivada, etc.

[245: gpvel1] Covarianza entre posición y velocidad: ejemplo (1)

**** 12:00

Materiales: [[CÓD.:](#) estimgradientofGPexES.mlx] [[PDF](#)]

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute un caso concreto de relación estadística entre ‘posición’ y ‘velocidad’ de un proceso gaussiano 1D. En este vídeo se revisan brevemente (simplemente se enuncian) las fórmulas teóricas de cálculo de covarianza entre posición y velocidad y autocovarianza de la velocidad. La discusión y demostración en detalle de dichas fórmulas se aborda en los vídeos [243] y [244].

Luego, se aplican dichas fórmulas y se grafican esa covarianza posición/velocidad y autocovarianza de velocidad, discutiendo brevemente su interpretación ‘física’. Su utilización para predicción se aborda en el vídeo [246].

[246: gpvel2] Covarianza entre posición y velocidad: ejemplo, predicción (2)

**** 17:48

Materiales: [[CÓD.:](#) estimgradientofGPexES.mlx] [[PDF](#)]

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo utiliza covarianzas posición/velocidad de un proceso gaussiano para hacer predicción de velocidad dadas medidas de posición en un ejemplo Matlab numérico. La segunda parte de dicho ejemplo discute cómo incorporar medidas mixtas de posición y velocidad para hacer predicción estadística de posición y velocidad (interpolación). Las varianzas y covarianzas propiamente dichas se calculan con las derivadas primeras y segundas del kernel de covarianza entre dos posiciones, según se detalla en el vídeo [245] cuya visualización previa se aconseja.

7.4.2 Ejemplo: el proceso (kernel) tipo Matérn

[247: maternEN] Matérn kernel and squared-exponential one: intuition as a filter with repeated real poles

**** 14:35

Materiales: [[CÓD.:](#) MaternKernelTST.mlx] [[PDF](#)]

[YouTube ►]

This video presents the relationship of the ‘Matérn’ type kernel with the result of filtering a continuous-time white noise with a filter (spectral factor) of type $F(s) = \frac{A}{(\tau \cdot s + 1)^m}$. It is verified that the power spectral density and autocovariance (Fourier inverse of the power density) coincide with the Wikipedia formulas based on the Gamma and Bessel functions for said Matérn kernel.

The limiting case (quadratic exponential kernel, Gaussian filter) is also discussed.

The study is done without demonstrations, based only on Matlab plots in time and frequency.

[248: maternr1EN] Matérn kernel regression (1): basic setup, 1st order filter example, linear interpolation

*** 16:40

Materiales: [[CÓD.:](#) SemipmaternReg.mlx] [[PDF](#)]

[YouTube ►]

This video presents an example of GP regression using a Gaussian process with autocovariance given by a Matérn kernel. The definition of it was discussed in the video [??], of which this is a continuation. To illustrate other concepts making the example richer, a 'parametric' component is incorporated saying that the stochastic process could not have zero mean and wishing to estimate that mean from a prior with a normal distribution, more details in other materials.

First, the example setup is presented and, then, a demonstration of interpolation with a spectral factor of order 1 is made. If the correlation distance ρ is small, with distant samples interpolation (mean) falls exponentially to the estimated "constant mean". If the correlation distance is large, then interpolation between nearby samples tends toward a piecewise linear interpolation.

Cases of higher order (and its relationship with splines), fractional order and infinite order (exponential-quadratic kernel) are discussed in the video [249], continuation of this one.

[249: maternr2EN] [Matérn kernel regression \(2\): 2nd 3rd and fractional order, splines, Gaussian filter examples](#)

17:25

Materiales: [[Cód.:](#) SemipmaternReg.mlx] [[PDF](#)]

[YouTube ▶]

This video, a continuation of [248], discusses the Matérn cases of spectral factor of order 2 and 3. When the correlation distance ρ increases, the interpolated mean tends to be a cubic spline (in case of order 2 spectral factor) and a quintic spline (in case of order 3), respectively.

The final part of the video presents an example of fractional order ($n = 1.5$) of the spectral factor (this motivates definitions of 'fractional derivatives' in literature, which are not the subject of this material), and an example with a quadratic exponential kernel, which is the Matérn limit when the order of the spectral factor tends to infinity (whose interpolation would be infinitely differentiable), as discussed in video [247].

7.4.3 Procesos estocásticos bidimensionales

[250: krig2dm] [Interpolación en procesos gaussianos 2D \(Kriging, Kernel Regression\): ejemplo Matlab](#)

**** 09:25

Materiales: [[Cód.:](#) regresionGaussian2inputs.mlx] [[PDF](#)]

[YouTube ▶]

Las fórmulas de estimación del vídeo [241] pueden aplicarse a variables índice t que sean multidimensionales, $t \in \mathbb{R}^2$ en vez de $t \in \mathbb{R}$ por ejemplo. Ello permite extender la idea de "filtrado en el tiempo" a "interpolación" en coordenadas espaciales (o espaciales y temporales en procesos, por ejemplo, gobernados por ecuaciones en derivadas parciales). Este vídeo presenta un ejemplo de interpolación 2D de una función a partir de unos puntos donde la "suavidad" de la función viene definida por la función de covarianza (*kernel*) especificada por el diseñador. Se usan las ecuaciones de mejor predicción y intervalo de confianza (proceso gaussiano) y se explora su interpretación en problemas de regresión/interpolación entendidos como filtrado estadístico. Las ideas de este vídeo están muy relacionadas con los "métodos Kernel" de regresión discutidos en otros materiales (por ejemplo, los vídeos [700], [701]); de hecho, el vídeo [702] sobre métodos denominados "regresión Kernel" enfatiza la relación prácticamente directa de dicha regresión con los conceptos de este vídeo.

[251: mimogp4predAEN] [\(4/5\) Multi-output Gaussian Processes: joint prediction](#)

17:42

Materiales: [[Cód.:](#) GPmultioutput.mlx] [[PDF](#)]

[YouTube ▶]


This video discusses prediction (Gaussian Process regression) in multi-output stochastic Gaussian processes, see video [211] for motivation and introduction.

Actually, with the representation as a process of a single output with an input domain expanded to $f(x, i)$ with i being a categorical ‘index’ or ‘label’ variable, prediction in an abstract sense is identical to the usual GP prediction, so there is nothing new from a theoretical point of view. Therefore, this video simply has a didactic objective: to understand how by measuring one variable another can be better estimated in close abscissa, complementing the case study of the videos [212] and [216].

If the ‘latent’ factors u in the statistical model had a physical meaning so its estimation seemed of interest in any particular application, that estimation is carried out in video [252], which closes the case study.

[252: mimogp5predUEN]

(5/5) Multi-output Gaussian Processes: latent factor estimation

 22:55

Materiales: [[Cód.:](#) GPmultioutput.mlx] [[PDF](#)]

[YouTube](#) ▶

This video discusses the prediction of the value of ‘internal’ hidden components or ‘latent factors’ of a 2-output Gaussian process where the statistical model is $y = Cu$, with y of size 2×1 being the observable outputs and u of 3×1 being the latent components, not directly ‘measurable’. Obviously, when the components are the ‘states’ of differential equations what we are doing would be analogous to Kalman filters or RTS/Wiener smoothing, but those techniques are not within the scope of this video.

Basically the covariance between the components and the outputs is $E[u(x_1)y(x_2)^T] = K(x_1, x_2)C^T$ so that using this covariance, the usual prediction formulas can be applied, illustrated in the video by a numerical example based on the case study of the previous videos, in particular [251].


7.5 Estimación a partir de EDO estocástica, ejemplo

En esta sección abordaremos un caso de estudio sobre estimar la velocidad de un sistema sujeto a aceleración aleatoria, a partir de su ‘representación en variables de estado’ $dx = Ax \cdot dt + G \cdot dB$ estocástica.

Comenzaremos con un filtro ‘naïve’ por ‘diferencia finita de posiciones’, pero el estimador óptimo con un histórico de muestras es el filtro de Kalman, como se dice en el último de los vídeos del caso de estudio; por su gran importancia teórica y tecnológica, dicho filtro se estudiará en detalle en el capítulo 17, aquí simplemente se utiliza una línea de Matlab para calcularlo y comprobar que es mejor que las propuestas ‘de memoria finita’ que sí detallamos.

[253: fdest1]

Estimación velocidad por diferencias finitas (1): motivación y planteamiento del problema

 12:41

Materiales: [[Cód.:](#) OnlyDIFinESP1.mlx] [[PDF](#)]

[YouTube](#) ▶

Este vídeo plantea el problema de estimar la velocidad (derivada) de una señal a partir de medidas de posición separadas un cierto intervalo de tiempo.


Realmente, es el primero de un caso de estudio de 4 vídeos. En este primer vídeo simplemente abordaremos el problema, planteándolo, discutiendo los conceptos de ruido de proceso (aceleración aleatoria), ruido de medida, la discretización del ruido de proceso, y la simulación para generar la señal que luego derivaremos aproximadamente con ‘diferencias finitas’.

El proceso en concreto será $1/(s + 0.25)^2$ sujeto a entrada ruido blanco de densidad espectral de potencia unidad. Se abordará el problema en representación de ‘ecuación diferencial estocástica’ $\frac{dx}{dt} = Ax + Gw$, o formalmente $dx = Axd t + Gdw$.

El detalle técnico de la estimación de la velocidad y su exactitud (o no) estudiada formalmente en términos estadísticos se abordará en el vídeo [254], siguiente de este caso de estudio.

[254: fdest2] **Estimación velocidad por diferencias finitas (2): análisis estadístico varianza error de predicción**

Materiales: [[CÓD.:](#) OnlyDiffFinESP1.mlx] [[PDF](#)]

***  14:48

[YouTube ▶]

Este vídeo continua el caso de estudio del vídeo [253], entrando ya en detalle técnico de cómo determinar el error al estimar la velocidad mediante la diferencia finita de dos posiciones.


Primero, calcula la covarianza entre estados (posición, velocidad) en un instante $x(0)$ y en el instante $x(T_s)$ siendo T_s el período de muestreo (separación entre muestras).

Con esa matriz de covarianzas, se hace un análisis de componentes principales (autovalores y autovectores) y se comprueba que, en efecto, existe una ecuación ‘casi determinista’ (varianza muy pequeña) que relaciona velocidades e incrementos de posición, en concreto $(pos(T_s) - pos(0))/T_s \approx (vel(0) + vel(T_s))/2$.

Si se desea usar posiciones para predecir velocidad ‘actual’ (no la media de velocidad actual y pasada), entonces se debe evaluar la varianza del error de predicción $\rho = (pos(T_s) - pos(0))/T_s - vel(T_s)$. Eso es lo que se realiza en la parte final del vídeo, primero sin ruido de medida y luego con ruido de medida. Se visualizan los resultados de los cálculos para dos valores del período de muestreo; el período de muestreo óptimo se discutirá en el siguiente vídeo [255] del caso de estudio.

[255: fdest3] **Estimación velocidad por diferencias finitas (3): comparación con estimador óptimo ventana finita**

Materiales: [[CÓD.:](#) KlmVsDiffFin.mlx]

**  15:19

[YouTube ▶]

Este vídeo continua el análisis estadístico de la predicción de velocidad por diferencias finitas en un proceso estocástico detallado en el vídeo [254].


Aquí se ejecutan los cálculos del anterior vídeo para un conjunto de valores del período de muestreo y se observa que la exactitud es mala a períodos muy pequeños y a períodos muy altos, pero que existe un intervalo de períodos de muestreo con una exactitud razonable.

La segunda parte del vídeo argumenta que las diferencias finitas $(p(T_s) - p(0))/T_s$ NO son la predicción óptima en sentido estadístico, sino que dicha predicción debe formularse a partir de la matriz de varianzas-covarianzas de las señales con las que se desea trabajar. Con ello se construye el predictor óptimo (de ‘memoria finita’ de 2 posiciones) y se compara con la diferencia finita ‘naive’. Obviamente, el óptimo estadístico tiene menor varianza.

Si el número de posiciones pasadas para calcular predicción de la velocidad tiende a infinito entonces el predictor óptimo es el *filtro de Kalman*, cuyo detalle y comparación se aborda en el vídeo [256] que cierra el caso de estudio.

[256: fdest4] **Estimación velocidad por diferencias finitas (4): comparación filtro Kalman**

Materiales: [[CÓD.:](#) KlmVsDiffFin.mlx]

****  15:19

[YouTube ▶]

Este vídeo concluye el caso de estudio de los vídeos [253], [254] y [255], donde se abordó la estimación de la velocidad por diferencia finita ‘naive’, y la estimación óptima de mínima varianza con dos medidas de posición.

Con infinitas medidas pasadas para estimar velocidad, el resultado es el filtro de Kalman como estimador óptimo (recurrente, memoria infinita). La teoría no es el objetivo aquí, simplemente se utilizan los comandos `dlqe` y `kalman` para calcular la varianza del error de predicción y comprobar que como era de esperar, es menor que la de todas las opciones discutidas en vídeos anteriores.

La parte final del vídeo discute que la suposición de ruido de medida de varianza constante conforme disminuye el período de muestreo deja de ser exacta cuando el período de muestreo se acerca a las constantes de tiempo de los filtros de las sondas y la electrónica de conversión analógico-digital, por lo que, aunque las gráficas parecen indicar que conforme el período de muestreo se reduce el filtro de Kalman converge a un estimado de velocidad de error ‘cero’, eso no puede ser cierto físicamente. Por brevedad, no se da detalle teórico de todo ello, remitiéndose al lector a otros materiales (pendientes).

7.6 Optimización Bayesiana

7.6.1 Motivación y esbozo de la metodología

[257: BOMot1] Optimización Bayesiana: introducción y motivación (parte 1)

Materiales: [BOIntroTeo.pdf]

**  16:47

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo introductorio plantea la necesidad de métodos estadísticos en optimización en algunos tipos de aplicaciones (optimización experimental, datos con ruido) donde no se dispone de un modelo de ecuaciones perfectamente conocidas a optimizar (eso caería bajo el marco de la optimización ‘determinista’, discutida en los primeros cinco minutos).

Un conjunto de algoritmos que usan modelos estadísticos en optimización se suele agrupar bajo el nombre de *optimización bayesiana* en la literatura. Este vídeo enuncia el problema, plantea el concepto de ‘función aleatoria’ y la necesidad de un kernel de covarianza para probar, entre otras cosas, continuidad de la función y, en general, introduce la idea de procesos estocásticos y procesos gaussianos como la herramienta básica que la mayoría de trabajos sobre optimización bayesiana utilizan.

Otras descripciones y aplicaciones de los procesos estocásticos se abordan en el vídeo introductorio [201]; futuros vídeos desarrollarán el problema de optimización bayesiana en mayor profundidad, continuando con [258].

[258: BOMot2] Optimización Bayesiana, introducción (parte 2): esbozo de la metodología

Materiales: [BOIntroTeo.pdf]

***  11:46

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este video continúa con la introducción y motivación a los problemas de optimización bayesiana que se inició en el video [257].

Aquí, revisamos dicho video en los primeros cuatro minutos, y luego discutimos el objetivo genérico de la optimización bayesiana: en realidad es un tipo de “diseño de experimentos” que intenta elegir muestras que tengan una alta probabilidad de darme un valor cercano al óptimo (explotación) o, bueno, tal vez deseemos muestrear los que actualmente probablemente sean subóptimos a cambio de que brinden mucha información sobre dónde estará el óptimo actual para muestras futuras (explorar).


Así, los algoritmos BO acaban dando lugar a los siguientes pasos:

- Configurar un modelo estadístico a priori
- Adquirir muestras (esto puede ser un procedimiento costoso en algunos casos de aplicación)
- Calcular un posterior (condicional a las medidas)
- Lleve a cabo un análisis estadístico del posterior para decidir la siguiente muestra y volver al paso 2, a menos que se cumpla algún criterio de terminación (número de muestras máximo, calidad de la última muestra, falta de progreso).

Por brevedad, un rápido resumen de los detalles de los pasos tercero y cuarto se pospondrá para el video [259], y los casos de aplicación y comentarios finales en el video [260] concluirán esta breve motivación y presentación del problema de optimización bayesiana. Un primer ejemplo numérico de la metodología se aborda en el vídeo [??].

[259: BOMot3] Optimización Bayesiana, introducción (parte 3): fórmulas del posterior y funciones de adquisición de próxima muestra (esbozo)

Materiales: [BOIntroTeo.pdf]

****  10:40

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este video continúa con la introducción y motivación a los problemas de optimización bayesiana que se inició en el video [257].

En el vídeo [258] concluimos que los algoritmos BO acaban comprendiendo los siguientes pasos:

- Configurar un modelo estadístico a priori
- Adquirir muestras (esto puede ser un procedimiento costoso en algunos casos de aplicación relevantes)
- Calcular un posterior (condicional a las medidas)
- Lleve a cabo un análisis estadístico del posterior para decidir la siguiente muestra y vaya al paso 2, a menos que se cumpla algún criterio de terminación.


Este video amplía el tercer paso de la metodología, argumentando que se lleva a cabo en un proceso gaussiano con las fórmulas condicionales de distribuciones normales multivariadas.

El último paso también se describe... la siguiente muestra puede seleccionarse para optimizar el valor esperado (EV), la probabilidad de mejorar (PI), la mejora esperada (EI), podemos ser arriesgados y elegir un límite de confianza inferior (nosotros probablemente no lo lograremos, pero obtendríamos una muy buena muestra si lo hiciéramos), u optimizar la ganancia de información mediante muestreo (búsqueda de entropía). Se presenta una descripción muy breve de cada una de estas opciones, sólo para vislumbrar las ideas principales, pero un análisis detallado será tema de otros materiales que analicen BO con mayor profundidad.

Un primer ejemplo numérico de la metodología se aborda en el vídeo [??]. Los casos de aplicación y las observaciones finales en el vídeo [260] concluirán esta breve presentación de los procedimientos de optimización bayesiana.

[260: BOMot4] Optimización Bayesiana, introducción (parte 4): aplicaciones, observaciones finales

Materiales: [BOIntroTeo.pdf]

**  17:57

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Después de una revisión rápida del material de vídeos anteriores, este video analiza los dominios de aplicación donde SÍ se recomienda la optimización bayesiana (optimización experimental sin modelos, parámetros de cálculos largos neuronales/fluidos/...), así como algunas circunstancias en las que NO se recomienda (funciones simples, casos de alta dimensión). Si no has visto los vídeos [257], [258] y [259], te recomiendo su visualización.

El vídeo concluye con un par de comentarios sobre, primero, el hecho de que el determinismo es un caso límite de estadística con un tamaño de incertidumbre que tiende a cero, por lo que hay bastantes problemas de optimización deterministas que admiten una "interpretación estadística" (por ejemplo, mínimos cuadrados, suponiendo un ruido de medición aleatorio normalmente distribuido); También existen opciones de optimización bayesiana semiparamétrica.

La segunda observación señala el hecho de que los métodos basados en gradiente pueden quedarse estancados en mínimos locales; por lo tanto, algunos algoritmos (búsqueda aleatoria dirigida) utilizan pasos de "tirar los dados" para intentar explorar y obtener el óptimo global; sin embargo, estas cuestiones no cambian el hecho de que el problema subyacente es determinista y no se supone que haya incertidumbre en la función objetivo.

Un resumen y conclusiones repasando las ideas de este vídeo y los anteriores de la serie que comenzó con el vídeo [257] finalizan el vídeo y el conjunto de 4 vídeos que motivan y describen brevemente las principales características de la optimización bayesiana.

[261: boloop1EN] Bayesian Optimization loop: a quick example of the methodology

Materiales: [[CÓD.:](#) BOIntroLOOPSimpleENG.mlx] [[PDF](#)]

*** 12:29

[YouTube ▶]

This video presents an example of the Bayesian Optimization methodology which we outlined in videos [??] and [??]. Code issues are only sparingly mentioned, as later videos will delve into the details and add some refinements.

The video focuses on understanding the Bayesian optimization loop (acquire a sample, update posterior, evaluate acquisition function for next sample), via a simple example using 'lower confidence bound' as said acquisition function heuristic.

7.6.2 Detalles, código y ejemplos

[262: boinopEN] Bayesian optimization: implicit information about the optimum in a Gaussian process

Materiales: [[CÓD.:](#) BOIntroOptimalSample.mlx] [[PDF](#)]

*** 18:53

[YouTube ▶]

This video plots the (approximate) distribution of global minima of the posterior of a function modeled as a Gaussian process of which three samples have been observed.

First, the video reviews how to obtain said posterior. Next, we review how to obtain realizations of it, and the code to repeat them and obtain the minimum of each one (this is the basic of the "Thompson sampling" heuristic in Bayesian optimization). With this code, six thousand minimums are obtained and three histograms are plotted:

- Marginal of x^* , probability density that in a given x^* there is a global minimum of some realization;
- Marginal of y^* , probability density that a certain y^* is the optimal value (global minimum) that achieves some realization;
- "joint" histogram approximating the joint density function over (x^*, y^*) representing the probability that some realization has the global minimum at x^* and its value is precisely y^* .

This is the implicitly available information about the location of the optimum. The goal of Bayesian optimization is to refine that information through sampling (explore) and/or to hit the optimum of my unknown underlying function as quickly as possible (exploit).

[263: boPIEN] Bayesian Optimization: probability of improvement, example

****  20:45

Materiales: [[CÓD.:](#) BOIntroAcquisitionShortEN.mlx] [[PDF](#)]

[YouTube ▶]

This video will discuss the meaning of some acquisition functions in Bayesian Optimization by means of an example. In particular, this first video will review the basic methodology, and it will set an example up with some mean, covariance kernel and dataset to work on it. If you are familiar with these concepts (if you already watched [??], [??] and [262]), you may skip to [10:30] when basic review and example setup have been already explained.

Then, the meaning of the ‘probability of improvement’ heuristic (PI) for BO acquisition function is discussed. In particular, it is shown to lie on the ‘conservative side’ if good samples (below the mean) are present: if I want to improve with almost certainty, I must descend a ‘little step’ in the downward direction (sort of gradient search), and also do not drift too far away from good samples to avoid uncertainty piling up. I will improve with large probability but, possibly, I will not jump quickly to the global minimum.

Other acquisition function options take a more ‘risky’ stance: they may not improve, but when they do improve, they may improve quite a lot more than the PI sample proposal.

[264: boEIEN] Bayesian Optimization: expected improvement, example (+LCB,PI,EV)

****  19:58

Materiales: [[CÓD.:](#) BOIntroAcquisitionShortEN.mlx] [[PDF](#)]

[YouTube ▶]

This video, a continuation of [263], will discuss the meaning of the *Expected Improvement* (EI) acquisition function in Bayesian optimization. Basically, Expected value (Gaussian mean) should be used if getting a bad sample penalises some “utility” in my particular application, and expected improvement should be used if bad samples do not incur any cost as we have our ‘best’ sample in our historical record to provide as our final solution to the optimization problem.

A final discussion on the comparison with probability of improvement and confidence bound acquisition functions is also provided.

All discussions are supported via an example that is coincident with that in the above-mentioned video this is a continuation thereof.

[265: болоop2EN] Bayesian optimization: PI, EI, LCB detailed example (Matlab)

**** 

15:59

Materiales: [[CÓD.:](#) BOIntroLOOPENG.mlx] [[PDF](#)]

[YouTube ▶]

This video presents an example of Bayesian optimization with probability of improvement, expected improvement, and confidence interval bound as acquisition functions. The development is similar to that of the video [261], but with greater detail in the presentation of the Matlab code.

[266: болоop3EN] Bayesian optimization: bad performance examples

***** 

17:17

Materiales: [[CÓD.:](#) BOIntroLOOPENG.mlx] [[PDF](#)]

[YouTube ▶]

This video continues video [265] with additional examples. The aforementioned video is reviewed in the first two minutes, and then the following examples are discussed:

- Example with one local minimum and the global minimum at an extreme of the search range (works OK)
- Example with “overexploitation” that proposes quite unreasonable samples in its final phase.
- Example with prior confidence interval too small: it gets stuck at a local minimum
- Example with too small a bandwidth of the prior (too large correlation distance): exploration is very bad, blindsided, and ends at a point that is not even a local minimum
- Example with too large prior bandwidth (too small correlation distance): it only extrapolates to a very short distance and, therefore, progress is very slow and, before approaching the true optimal zone, it explores in an almost regular mesh (at the distance where correlation vanishes).

With all these examples, we seek to depict a complete image of the advantages and disadvantages of Bayesian optimization, and the importance of a good choice of the prior... or, if the prior is not so good, detect overexploitation and low probability of the samples (given the prior) to adjust the (hyper)parameters of said prior as samples are acquired. These issues are not considered here.

Capítulo 8

Sistemas multivariables: análisis de propiedades

8.1 Causalidad y realizabilidad

[267: cau] Realizabilidad y Causalidad

Materiales: [Causalidad.pdf]

**  11:29
[YouTube ►]

Este vídeo describe el concepto de realizabilidad y el de causalidad (sistema “no anticipativo”). Estas nociones ya se han introducido desde un punto de vista genérico de teoría de sistemas, en los vídeos [9] y [10]: en los sistemas realizables, toda la historia “pasada” debe poder resumirse en un estado “presente”.

Aquí se abordan consideraciones particulares de sistemas lineales en función de transferencia o representación interna.

En concreto, en este vídeo, se presentan los sistemas continuos realizables (y causales) como aquéllos cuya función de transferencia $G(s)$ es propia (grado numerador \leq grado denominador). La misma conclusión es obtenida (grado numerador \leq grado denominador) en el caso discreto $G(z)$.


El concepto de causalidad está íntimamente ligado a la existencia de una representación en variables de estado, dado que los integradores son realizables con componentes físicos (acumulación de caudales, cargas en condensadores, ...) mientras que los derivadores ideales no (respuesta a alta frecuencia infinita).

Se presenta un ejemplo de filtro discreto no causal. Los filtros no causales podrían ser usados en aplicaciones fuera de línea o con “buffer”. Aunque estos filtros resultan bastante “intuitivos”, la fundamentación teórica de los filtros no causales se basa en la transformada de Fourier, ver vídeo [198]. Los filtros no causales en representación interna se suelen realizar siguiendo las fórmulas de Rauch-Tung-Striebel, ver vídeo [607].

8.2 Estabilidad, ganancia, polos y ceros, frecuencia

[268: mdtvss] Sistemas multivariables: ventajas e inconvenientes de matriz de transferencia versus representación en variables de estado

Materiales: [MdTPart2.pdf]

***  10:58
[YouTube ►]

Este vídeo discute las principales ventajas e inconvenientes de utilizar la representación en Matriz de Transferencia (vídeo [131]) versus usar la representación interna (variables de estado).

Las ventajas de trabajar con funciones/matrices de transferencia son:

1. Evitar derivadas o integrales en la manipulación de ecuaciones
2. Interpretación del efecto conjunto de las entradas como *superposición*
3. Interpretación de cada efecto individual
4. Identificación experimental de cada efecto individual
5. Existencia de estrategias de control *descentralizado* que utilizan cada elemento (por separado) de la matriz de transferencia, muy comunes en industria.

Las desventajas de trabajar en el dominio de Laplace son:

1. Mayor lejanía de la “física”: la representación interna codifica las ecuaciones de la física como vienen en los libros de texto de Física (ecs. diferenciales, balances de fuerzas, energías, ...), sin ningún paso ‘abstracto’ a Transformada de Laplace.
2. Mayor robustez numérica de la representación en variables de estado lineal (dado que (A, B, C, D) son matrices constantes mientras que $G(s)$ es una matriz cuyos elementos son cocientes de polinomios).
3. La representación $G(s)$ sólo aplica a sistemas lineales, mientras que $dx/dt = f(x, u)$ puede ser no lineal.

[269: propsmot]

Análisis de propiedades en sistemas multivariables: motivación
y planteamiento del problema
Materiales: [analisipropiedadesIntro.pdf]

**
 07:44
[YouTube ▶]

Este vídeo motiva la necesidad del análisis de propiedades presentando las fases de un diseño de control multivariable: modelado, análisis de propiedades, y si no son satisfactorias, selección de instrumentación y control. Obviamente, las preguntas aplican tanto a casos ‘multivariables’ como ‘monovariabes’ (1 entrada, 1 salida; cursos introductorios). Responderlas será más complicado en el caso multivariable, claro.

El resumen es que, cuando se dispone de un modelo (en fase de proyecto) o de un prototipo (en fase de pruebas) de un sistema tecnológico, es necesario discutir:

- puntos de equilibrio nominales del sistema no lineal, puntos de operación
- estabilidad del punto de operación deseado,
- cómo cambian los puntos de equilibrio cuando algunas entradas se incrementan (ganancia),
- tiempo de establecimiento (cuánto se tarda en ir de un punto de operación a otro, cuánto tarda el sistema en recuperarse de una modificación de condiciones de operación)
- Presencia o no de oscilaciones/sobrepasamiento en el transitorio
- En algunos problemas (vibraciones, corriente alterna, ...), respuesta ante entradas periódicas (respuesta en frecuencia).

[270: propsest]

Análisis de propiedades de sistemas multivariables lineales:
estabilidad
Materiales: [3analisipropiedades2.pdf]

 08:26
[YouTube ▶]

Este vídeo discute brevemente la estabilidad de sistemas lineales multivariables: parte real polos continuos negativa, o polos discretos de módulo menor a 1. Los resultados son prácticamente idénticos a bien conocidos resultados de sistemas monovariantes.

Se recomienda al alumno revisar los conceptos de valores singulares (Sección A.4, como mínimo los vídeos [svdOA(11:35)] y [svd2(17:10)]), para la comprensión correcta del video [propsgan(10:00)] a continuación.

[271: propsgan] Análisis de propiedades en sistemas multivariables lineales:
matriz de ganancia estática

Materiales: [3analispropiedades2.pdf]

**** 10:00

[YouTube ▶]

Este vídeo discute el concepto de matriz de ganancia estática (la relación entre ganancia y teorema del valor final en Laplace se estudia en el vídeo [134]). Su fórmula es $G(0)$ en el caso continuo y $G(1)$ en el caso discreto, siendo G el modelo en función o matriz de transferencia (introducción al concepto de matriz de transferencia en vídeo [131]) y, en representación interna $C(-A)^{-1}B + D$ (continuo) y $C(I - A)^{-1}B + D$ (discreto).

Se motiva la utilidad de los conceptos de ganancia máxima y mínima y su obtención basada en valores singulares (valores propios de una matriz por su transpuesta). Se hace un ejemplo numérico sencillo para ilustrar la idea.

[272: propsresto] Análisis de propiedades en sistemas multivariables: tiempo
establecimiento, resp. frecuencia, ceros, diag. bloques

12:48

Materiales: [3analispropiedades2.pdf]

[YouTube ▶]

Una vez discutidos conceptos de estabilidad y ganancia estática, este video presenta brevemente toda una serie de propiedades adicionales que permitan caracterizar el comportamiento dinámico de un sistema multivariable:

- tiempo de establecimiento (fórmulas aproximadas de 1er orden, casos continuo y discreto)
- Matriz de respuesta en frecuencia, y ganancias máxima y mínima en función de la frecuencia, valores singulares $\sigma(j\omega)$
- Normas 2 e infinito de un sistema dinámico (esbozo breve, detalle en video [633] y ejemplos Matlab en vídeos [634], [635]).
- Ceros de transmisión de un sistema en representación interna
- Propiedades en bucle cerrado: necesaria manipulación de diagramas de bloques matriciales.

Los valores y vectores propios de la matriz A de la representación en variables de estado permiten un análisis en detalle de los “modos” de la respuesta libre; se recomienda visualizar el vídeo [171] para comprender adecuadamente esa idea sobre un ejemplo de sistema mecánico de 3 masas y 4 muelles.

8.2.1 Ejemplos Matlab

[273: tuboml] Modelado y análisis de propiedades de un sistema térmico
(tubería corta), ejemplo Matlab

Materiales: [CÓD.: ModeloTuboTermicoDeterminista.mlx] [PDF]

*** 09:50

[YouTube ▶]

En este vídeo se modela una tubería por la que circula un gas a una cierta temperatura, que calienta la tubería. También se incluye una sonda de temperatura fijada a dicha tubería, con su propia dinámica.

Para evitar modelos en derivadas parciales, se supone una tubería “corta” de modo que la energía extraída del gas para calentar la tubería se considere despreciable y la temperatura constante a lo largo de la tubería. Con ello, el sistema tendrá dos entradas T_{gas} y $T_{ambiente}$, y dos variables de estado T_{tubo} , T_{sonda} .

El vídeo justifica las ecuaciones de la representación interna y construye el modelo (`ss`), presenta su función de transferencia (`zpk`), su matriz de ganancia (`dcgain`) y respuestas ante escalón (`step`) y en frecuencia (sólo magnitud, `bodemag`). Se comprueba la relación aproximada entre el ancho de banda (semiperíodo asociado) y el tiempo de establecimiento (nota: en sistemas complejos, con polos oscilatorios la relación aproximada es que el tiempo de *subida* es inversamente proporcional al ancho de banda).

[274: termem] Sistema térmico orden 4: Análisis de propiedades, resp. temporal y en frecuencia (Matlab)

Materiales: [[CÓD.:](#) termexampleModelo.mlx] [[PDF](#)]

*** 10:58

[YouTube ▶]

Este vídeo implementa en Matlab con el comando `ss(A,B,C,D)` el modelo de sistema térmico de orden 4 cuya obtención se detalla en el vídeo [18], resumida aquí en el primer minuto.

Con dicho modelo, se analizan las propiedades en equilibrio (matriz de ganancia estática), y se calcula un punto de funcionamiento de actuadores para compensar diferencias de temperatura de 15 grados entre el interior y el exterior.

Se examinan:

- la respuesta ante escalón (`step`)
- los modos de la respuesta libre (valores y vectores propios de A); un ejemplo complementario de análisis de dichos modos sobre un sistema oscilatorio (modos de vibración) puede verse en el vídeo [171].
- El diagrama de Bode (`bodemag`)
- El diagrama de valores singulares de $G(j\omega)$ en frecuencia (`sigma`).

El dimensionamiento de actuadores para conseguir determinadas prestaciones de control sin saturación en este modelo se discute en los vídeos [439] y [440].

[275: props2mass] Análisis de propiedades de la resp. temporal de un sistema de 2 masas y 2 muelles (Matlab)

Materiales: [[CÓD.:](#) propiedadesmuelle.mlx] [[PDF](#)]

*** 10:25

[YouTube ▶]

Este vídeo discute las propiedades de la respuesta temporal del sistema de dos masas, muelles y amortiguadores cuyo modelo se discutió en el vídeo [88]; un resumen muy breve de dicho modelado se aborda en los dos primeros minutos aquí.

Para comprobar que la discretización no altera dichas propiedades, el modelo es también discretizado con retenedor de orden cero (`c2d` , vídeo [371]).

Se analiza primero la *estabilidad* comprobando la parte real (continuo) o el módulo (discreto) de los polos.

Una vez comprobado que el sistema es estable, se obtiene la matriz de ganancia estática. Se realiza la descomposición en valores singulares de dicha matriz con `svd` , y se interpretan físicamente las direcciones de entrada y salida asociadas a las ganancias máxima y mínima.

Por último, se usan fórmulas de tiempo de establecimiento para estimarlo.


El análisis de propiedades, finalmente, es verificado obteniendo la respuesta ante escalón del sistema con **step**, así como la respuesta ante un escalón simultáneo en las dos entradas asociado con la maniobra de menor ganancia estática.

Por brevedad, las propiedades en frecuencia de este sistema y sus ceros se analizan en el vídeo [276].

Los autovectores y autovalores de la matriz A de la representación interna permiten un análisis más detallado de los modos de la respuesta libre; un análisis de un sistema mecánico muy similar a este se aborda en el vídeo [171].

[276: freq2mass]

Análisis de propiedades de un sistema de 2 masas y 2 muelles
(Matlab): respuesta en frecuencia y ceros


07:01

Materiales: [[CÓD.:](#) propiedadesmuelle.mlx] [[PDF](#)]

[\[YouTube ▶\]](#)

Este vídeo discute las propiedades de la respuesta del sistema de dos masas, muelles y amortiguadores cuyo modelo se discutió en el vídeo [88], continuando el análisis iniciado en el vídeo [275].


En este vídeo se analiza la respuesta en frecuencia del sistema, tanto a título individual (cada elemento de la matriz de transferencia de forma aislada, con **bode**) como de forma conjunta con el diagrama de valores singulares en función de la frecuencia, **sigma**.

Tras ver el diagrama de respuesta en frecuencia, se calculan las normas 2 e ∞ del sistema, y se discute la interpretación de dichos resultados. Opcionalmente, podría ser de interés en este momento visionar ejemplos Matlab adicionales de los vídeos [634] y [635] para comprender mejor estas ideas y también el significado de la respuesta en frecuencia multivariable.

La parte final del vídeo aborda los ceros de transmisión (no existe ninguno) y los ceros de las funciones de transferencia individuales considerados de forma aislada.

[277: bc2mass]

Análisis de propiedades de un sistema de 2 masas y 2 muelles
(Matlab): bucle cerrado


14:58

Materiales: [[CÓD.:](#) propiedadesmuelle.mlx] [[PDF](#)]

[\[YouTube ▶\]](#)

Este vídeo propone un controlador proporcional diagonal para el sistema de dos masas, muelles y amortiguadores cuyo modelo se discutió en el vídeo [88] y cuyas propiedades temporales y frecuenciales en bucle abierto se habían analizado en los vídeos [275] y [277].

Se analizan los polos, ganancia, svd, sigma y respuesta ante escalón de bucle cerrado con dicho control proporcional.

En la segunda mitad del vídeo, se añade una acción integral, comprobándose que el error en régimen permanente se hace cero, y analizando los cambios en las respuestas temporal y en frecuencia.

Sistemas de parámetros distribuidos

Los vídeos [props2mass(10:25)], [freq2mass(07:01)] presentaban el análisis de un sistema de orden 4. En ocasiones, se debe analizar las propiedades de un sistema de orden elevado proveniente de, por ejemplo, un modelo de elementos finitos mecánico, térmico, etc. Aquí tenemos un ejemplo con sistemas que llegan a tener orden 80:

[278: fesim]

Simulación de modos de vibración longitudinales de un muelle
mediante modelo de orden elevado


10:51

Materiales: [[CÓD.:](#) SimulaMuelleOndaLongitudinal.mlx] [[PDF](#)]

[\[YouTube ▶\]](#)

Este vídeo simula un ejemplo numérico concreto de las vibraciones de un muelle con un extremo fijo, cuyo modelado se ha considerado en el vídeo [78] en detalle, como ejemplo de modelos en parámetros distribuidos y elementos finitos. La parte inicial revisa de forma muy rápida el modelado, para hacer a este vídeo [278] relativamente autocontenido. A partir del minuto 2, se comparan las respuestas temporal (ante escalón), en frecuencia, y los polos y ceros de los modelos con uno, diez, veinticinco y cuarenta elementos que modelan el muelle. Como se observa que subir de 25 a 40 apenas cambia la respuesta, se decide no aumentar más el número de elementos, generando un sistema dinámico lineal de orden 80.

La respuesta que se había comparado era la relación entre fuerza y desplazamiento del extremo libre del muelle. Para comprobar la relevancia de los distintos modos de vibración, la parte final del vídeo simula ante una entrada senoidal la posición de los 40 elementos del muelle, cambiando la ecuación de salida. A frecuencias bajas se observa que los desplazamientos son proporcionales a la distancia al extremo fijo (muelle sin dinámica interna, deformación constante a lo largo de toda su longitud), y que la relación fuerza-desplazamiento es prácticamente igual a $F = kx$. A frecuencias a partir de la primera resonancia ya se observan claramente distintos modos de vibración que hacen que $F = kx$ ya no sea una descripción adecuada. Tampoco incorporar fricción con $F = (k + f \cdot s)x$ es una descripción adecuada, dado que no predice las resonancias observadas debido a las ondas longitudinales.

Nota: En aplicaciones no es necesario modelar un muelle-amortiguador como un sistema de orden 80. Este vídeo únicamente analiza las propiedades de dicho modelo, pero una vez discutidas ciertas nociones de controlabilidad y observabilidad de los estados, podrá plantearse la eliminación de los modos de vibración que requieran de una fuerza de elevada amplitud y frecuencia para ser excitados. Este muelle en concreto se abordará en los vídeos [297] y [298], con distintas opciones teóricas detalladas en otros materiales, cuya comprensión es muy importante: modelar un muelle como se ha hecho en este vídeo raramente tiene justificación práctica, como se demuestra en dichos vídeos, salvo que las fuerzas que vayan a actuar sobre él sean de muy alta frecuencia y su masa sea comparable al resto de masas del sistema mecánico completo del que pudiera formar parte.

Otro ejemplo de modelado con elementos finitos de un calentador tubular aparece en el vídeo [187].

8.3 Controlabilidad y Observabilidad

[279: ctrb]
Controlabilidad de sistemas dinámicos lineales (tiempo discreto) *****/ ****

28:34

Materiales:
[ControllObserv.pdf]

[PLAY ▶]

Este vídeo es antiguo, largo, y tiene algunas erratas/incorrecciones. Está en proceso de revisión


Planteamiento del problema de controlabilidad y observabilidad.

*Nota: los estados “no controlables” pueden aparecer durante la manipulación de las ecuaciones algebraico-diferenciales de los modelos de sistemas físicos. Se aconseja visionar el vídeo [30] para entender mejor la relevancia de este problema en las fases de modelado. También puede haber problemas de controlabilidad y observabilidad debidos a cancelaciones entre polos de controlador y ceros de proceso, o ceros de controlador y polos de proceso, ver vídeo [287].


Caso estático: estimado mínimo cuadrático de x dado y ante $y = Cx$; elipsoide alcanzable en $x = Cu$ ante entrada u acotada.


Revisión de teorema de Cayley-Hamilton, y de los conceptos de espacio columna y espacio nulo de una matriz.

Controlabilidad discreta (horizonte finito, entrada determinista), matriz de controlabilidad.

Horizonte infinito: gramiano de controlabilidad. Ejes de elipsoide alcanzable, valores propios del gramiano (grado de controlabilidad). 


Ejemplo (código Matlab) de análisis de controlabilidad, sobre el sistema del video [275].


Interpretación estadística (prerequisitos en video [219]): el gramiano de controlabilidad como matriz de varianzas covarianzas estacionaria ante entrada ruido blanco. Elipsoide de confianza de encontrar al estado.
***** 


[280: obsv] Observabilidad de sistemas dinámicos lineales en representación interna ***** / *****  14:32
Materiales: [ControllObserv.pdf] [PLAY ►]

Planteamiento del problema.

Estimación de estado a partir de una secuencia finita de muestras (caso lineal discreto). Matriz de observabilidad.

Horizonte infinito: gramiano de observabilidad. Valores y vectores propios del gramiano: interpretación. 

Interpretación estadística del gramiano de observabilidad. ***** 

[281: obsvml] Observabilidad de sistemas en representación interna: ejemplo Matlab *****  12:57
Materiales: [Cód.: ctrbobsvex.mlx] [PDF] [YouTube ►]

En este vídeo se analizan, con el adecuado código Matlab, las propiedades de observabilidad sobre el sistema del video [275] (su controlabilidad se analizó en el video [279]). Se normaliza el ruido, y se analizan, mediante los autovectores y autovalores del gramiano de observabilidad, las combinaciones de estado más fáciles o difíciles de estimar. Se analizan varias combinaciones de sensores o períodos de muestreo en términos de observabilidad.

Un ejemplo más sofisticado de análisis de observabilidad y controlabilidad en un bucle cerrado se aborda en el video [287].

[282: canf] Controlabilidad y observabilidad: consideraciones finales *****  08:53
Materiales: [ControllObserv.pdf] [PLAY ►]

En este video se esboza el análisis de controlabilidad y observabilidad para el caso de sistemas de tiempo continuo.

También se presenta la descomposición en forma de Kalman (controlables y observables, controlables no observables, observables no controlables, y no controlables no observables), que motiva la realización mínima (que calcula el comando **minreal** en Matlab), y las definiciones de sistemas estabilizables/detectables.

Nota: El concepto de controlabilidad y observabilidad está íntimamente relacionado con el concepto de *estabilidad interna* que es de vital importancia. Se recomienda al lector revisar la Sección 8.4.1 y los vídeos allí contenidos.

8.4 Manipulación e interconexión de sistemas

[283: opss] Operaciones en representación interna: selección de subsistemas, conexión en cascada y en suma

Materiales: [operSS1.pdf]

*** 11:00

[YouTube ►]

En este vídeo se estudia, al principio del mismo, la manipulación más trivial de un sistema, que es la mera selección de un subconjunto de variables (entradas y salidas) de un modelo multivariable complejo, motivada, por ejemplo, por la necesidad de analizar posteriormente las propiedades de ese “subsistema” con un número reducido de entradas y salidas (en principio, el número de estados se mantiene sin cambios, salvo que algunos dejen de ser “controlables” u “observables” al eliminar entradas o sensores).

A continuación, se detalla cómo conectar dos sistemas (posiblemente multivariables), descritos en representación interna (variables de estado). En concreto, se estudian las conexiones en serie (la entrada de un sistema es la salida de otro) y en suma (se suman las salidas), como los ejemplos más sencillos de este tipo de manipulaciones.

[284: invss] Inversa de sistemas dinámicos en representación interna

Materiales: [SSinversa.pdf]

*** / **** 08:36

[YouTube ►]

Este vídeo discute cómo calcular la inversa de un sistema $G = ss(A, B, C, D)$, demostrando que, si D , es invertible dicha inversa G^{-1} es el sistema

$$G_{inverso} = ss(A - BD^{-1}C, BD^{-1}, -D^{-1}C, D^{-1})$$

Si D no fuera invertible, la inversa, que sí existiría en matriz de transferencia (p.ej. la inversa de $1/(s+1)$ es $(s+1)$), no sería *realizable* como una representación interna normalizada, según discutido en el vídeo [267] y, por tanto, no podría expresarse como $ss(\cdot, \cdot, \cdot, \cdot)$.

También se demuestra, utilizando el denominado lema de Schur sobre determinante de matrices particionadas en bloques, que los polos de G son los ceros de G^{-1} y los ceros de G son los polos de G^{-1} , como era de esperar intuitivamente. ****

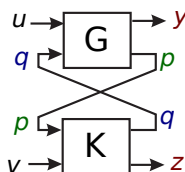
[285: starp] Interconexión genérica realimentada de sistemas dinámicos (star product), fórmulas en representación interna

Materiales: [starP.pdf]

**** 18:21

[YouTube ►]

Este vídeo plantea cómo obtener las ecuaciones de bucle cerrado de una interconexión entre dos sistemas multivariables dada por el diagrama de bloques:



para el caso de que G y K estén disponibles en representación interna normalizada. En el desarrollo, las matrices de las representaciones internas se particionan adecuadamente con dimensiones compatibles a cada grupo de entradas y salidas. El resultado es un sistema en bucle cerrado cuyas entradas son (u, v) y cuyas salidas son (y, z) , habiendo eliminado p y q , aunque realmente en las manipulaciones intermedias, se obtiene su ecuación de salida involucrando a los estados x_G y x_K y las entradas (u, v) , por lo que podría ser utilizada si así se deseara.

Tras resolver y obtener la fórmula de interconexión, se constata que es necesario que determinada matriz sea invertible. Ello, por tanto, requiere definir qué se entiende por una interconexión “bien planteada” precisamente en términos de invertibilidad de dicha matriz. El problema puede manifestarse cuando existen bucles algebraicos entre p y q .

Se pone un ejemplo de un bucle cerrado que NO está bien planteado y, por tanto, da lugar a matrices de transferencia entre algunas de las variables que NO son realizables y, por tanto, no admiten una representación interna con las fórmulas antes desarrolladas. Esta situación de bucles mal planteados debería ser evitada en la práctica.

El ejemplo de Matlab en la segunda mitad del vídeo [286] también ilustra esta interconexión y los problemas que pueden ocurrir al interconectar procesos inestables con procesos de fase no mínima.


8.4.1 Estabilidad interna, controlabilidad y observabilidad en bucle cerrado

Esta sección discute el concepto de “estabilidad interna”. La idea básica viene motivada porque pueden existir sistemas con modos no controlables, que si fueran inestables, imposibilitarían el funcionamiento físico del sistema (no puede evitarse que vayan a infinito); también pueden existir sistemas con modos no observables, que si fueran inestables, indican que con los sensores presentes no nos daríamos cuenta de dicha inestabilidad y, por tanto, el sistema físico tampoco podría hacerse funcionar manteniéndose cerca del punto de operación prefijado.

Los sistemas donde **no** existen modos “no controlables inestables” (o sea, que todos los modos no controlables, si los hubiese, tienen dinámica estable) se denominan *estabilizables*. Aquellos donde **no** existen modos “no observables inestables” (o sea, que todos los modos no observables, si los hubiese, son estables) se denominan *detectables*. Por tanto, sólo pueden funcionar correctamente alrededor de un punto de operación los sistemas físicos **estabilizables y detectables**. En un sistema estabilizable y detectable, estabilidad “interna” (matriz A con polos estables) es “equivalente” a estabilidad entrada-salida (todos los elementos de la matriz de transferencia, después de simplificar polos y ceros idénticos, son estables).

[286: estint] [Interconexión de sistemas \(representación en variables de estado\): estabilidad interna](#)

Materiales: [estabinterna.pdf]

****  08:09

[YouTube ▶]

Este vídeo discute el concepto de *estabilidad interna* de una interconexión de sistemas, en concreto de la interconexión realimentada genérica (star-product, [lft](#)) discutida en el vídeo [285].

En particular, se discute que la interconexión de sistemas (incluso cuando, de forma separada, sean estabilizables y detectables), puede dar lugar a bucles cerrados que no sean *estabilizables y detectables*. Ello ocurre si en la interconexión se produce una cancelación polo-cero inestable. El resultado es un sistema *internamente inestable*, en el que las funciones de transferencia resultantes pueden ser estables pero haber estados internos de alguno de los subsistemas que vayan a infinito (y no se vean en la FdT por no ser controlables u observables).

Para ilustrar los conceptos, se desarrolla un ejemplo en Matlab (utilizando los comandos [ss](#), [lft](#), y [minreal](#)) para calcular la representación interna y función de transferencia de un bucle que *no* es internamente estable.

Incluso con cancelaciones de polos y ceros *estables* pueden aparecer problemas tecnológicos, como discute el vídeo [287] en detalle (el vídeo [400] plantea ejemplos similares de modo más sencillo, sin entrar en controlabilidad y observabilidad).

[287: cancobsctr]
18:10

Controlabilidad y observabilidad bucle cerrado: ejemplo Matlab
regulador de cancelación (IMC discreto)

[YouTube ▶]

Materiales: [[CÓD.:](#) testIMCcancel.mlx] [[PDF](#)]

En este vídeo se busca analizar un bucle cerrado de control en términos de controlabilidad y observabilidad. Se trabajará en tiempo discreto. Primero, se diseña un controlador basado en “cancelación”, en concreto control por modelo interno (IMC), y se representará su respuesta temporal y en frecuencia en bucle cerrado ante entradas referencia y perturbación a la entrada, siendo las salidas las variables manipulada y controlada. El diseño es, intencionalmente, incorrecto, pero no es objetivo detallar todo esto en este material: para más detalles sobre diseños IMC discreto, puede verse, si así se desea, el vídeo [419].

Una vez diseñado el controlador, se observan comportamientos poco amortiguados en tiempo y grandes picos de resonancia en frecuencia.

El análisis de controlabilidad y observabilidad con 2 entradas y 2 salidas da correcto (todo controlable y observable): eso significa que ninguna otra entrada, medida o condición inicial nos va a dar “sorpresas”, eso es bueno.

Si analizamos controlabilidad y observabilidad sólo ante una de las dos entradas o ante una de las dos salidas, entonces se observan polos no controlables o no observables que, al estar poco amortiguados, producen los efectos no deseados en la respuesta temporal y frecuencial.

La última parte del vídeo forma la forma canónica *modal* del bucle cerrado, para observar claramente qué polos se controlan/observan con qué entradas y salidas, interpretando adecuadamente su significado y la relación con la respuesta temporal antes simulada.

Otros controladores de cancelación como *mixsyn* también podrían tener los mismos inconvenientes que lo aquí discutido, ver vídeo [821].

En resumen, no sólo las cancelaciones polo-cero inestables o fase no mínima (vídeo [286]) causan problemas, sino que aquéllas que se “acercan” a esa frontera de la estabilidad también pueden ser estables “matemáticamente” pero tener problemas en respuesta temporal/frecuencial que impidan su aplicación tecnológica en una aplicación.

Capítulo 9

Aproximación de orden reducido de sistemas lineales

En sistemas monovariantes, el concepto de reducción de orden está muy relacionado con los conceptos de dominancia o cancelación polo/cero, discutidos en el abundante material de terceros para el caso monovariante. El objetivo de este capítulo es generalizar adecuadamente las ideas a casos multivariantes.

9.1 Caso monovariante, ideas básicas

[288: domycanc] Dominancia y Cancelación

controltheoryorg (canal YouTube)

**  16:27
[PLAY ►]

Este vídeo discute los conceptos de dominancia y cancelación para obtener modelos simplificados (orden reducido) de funciones de transferencia *estables*.

- La dominancia la propone como “distancia al origen”: escogiendo el polo (o par de polos) con menor parte real (*polos dominantes*), el autor propone trazar una circunferencia alrededor del origen de 5 veces el módulo del mismo y eliminar todos aquéllos polos fuera de dicho círculo (otras referencias proponen 10 veces en vez de 5, a veces la elección depende del orden final deseado y la precisión deseada).
- La cancelación se produce cuando un polo está “cerca” de un cero (que originaría un término muy pequeño en el numerador –residuo asociado– de la descomposición en fracciones simples, que podría despreciarse). El concepto de “cerca” depende de la distancia al origen: si los polos están cerca del origen deben estar más “pegados” que si están lejos. La parte final del vídeo discute la cancelación de polos/ceros inestables, que como el autor argumente, NO debe realizarse.

En ambos casos (eliminación polos no dominantes o casi cancelados), debe conservarse la ganancia estática.

[289: ordnafopd] Aproximación de sistemas lineales de orden superior a primer/segundo orden + retardo: enfoque serie de Taylor
Materiales: [[CÓD.](#): PrimerOrdenyRetardoAprox.mlx] [[PDF](#)]

***  10:25
[YouTube ►]

En este vídeo se plantea cómo reducir un proceso de orden superior, en concreto $G(s) = \frac{3(-0.15s+1)e^{-0.06s}}{(1.1s+1)(0.5s+1)(0.1s+1)^3}$ a un modelo de primer orden + retardo o de segundo orden + retardo.

La motivación es debido a que los modelos de primer/segundo orden + retardo son habituales como modelos de procesos (fáciles de identificar ante un experimento escalón, vídeos [154], [748]). También existen técnicas de ajuste de PID's óptimos o via IMC (vídeos [403], [405]) para esa clase de procesos.

La idea básica es la serie de Taylor... hasta grado 1 (término s de dicha serie) podemos aproximar $e^{-\tau s} \approx \frac{1}{\tau s + 1} \approx 1 - \tau s$. Con ello, se puede cambiar el cero $(-0.15s + 1)$ por un retardo de 0.15 y los polos $(0.5s + 1)(0.1s + 1)^3$ por un retardo de 0.8. Si se desea un modelo de segundo orden + retardo, sólo habría que cambiar el cero $(-0.15s + 1)$ por un retardo de 0.15 y los polos $(0.1s + 1)^3$ por un retardo de 0.3.

Si se admite modificar las constantes de tiempo del modelo resultante (que no coincidiera con la del polo dominante, esto es, 1.1), entonces se puede lograr la coincidencia de términos hasta grado *dos* (s^2) del desarrollo en serie de Taylor de $G(s)$ y de $\frac{1}{\tau s + 1} e^{-ds}$.

El uso de la serie de Taylor implica que la aproximación es válida cerca de $s = 0$, o sea, cerca de $\omega = 0$ (frecuencias bajas) si consideramos la aproximación en el dominio de la frecuencia.

La coincidencia de las series de Taylor hasta un cierto orden entre una función (a aproximar) y otra función parametrizada de diferente estructura (por ejemplo, cociente de polinomios) es la idea fundamental que está detrás de la aproximación de Padé. En el caso del retardo, se aborda en el vídeo [191].

[290: domp1] Dominancia (polos): idea básica y ejemplo Matlab

***  18:57
[YouTube ▶]

Este vídeo discute el concepto de 'polo dominante' basándose en el sistema $G(s) = 3/(s + 1) \cdot p/(s + p)$, con un polo en -1 y otro en $-p$.

La descomposición en fracciones simples de la respuesta escalón $G(s) \cdot 1/s$ indica que si p es mucho mayor que 1, entonces el término asociado a e^{-pt} puede despreciarse porque el coeficiente que lo multiplica es pequeño y, además, sea o no pequeño, e^{-pt} cae a cero mucho más rápidamente que e^{-t} , con lo que, a partir de $3/p$ aproximadamente, sólo será visible en la respuesta e^{-t} + término constante, con lo que llamaremos a e^{-t} la exponencial *dominante* y a e^{-pt} la exponencial 'dominada'... o 'polo dominante en -1 ' y 'polo dominado en $-p$ '.

Con ello se puede describir de forma aproximada con una exponencial (sistema de 1er orden $3/(s + 1)$) la respuesta ante escalón cuando $p > 10$.

La segunda parte del vídeo demuestra que el concepto de dominancia sólo aplica cuando se desea simplificar la descripción de la dinámica antes señales *lentas*: si las señales tienen armónicos importantes cuyo período es 'comparable' a $3/p$ o menor, o pulsos cuya duración es 'comparable' a $3/p$ o menor, entonces NO aplica. EL vídeo presenta ejemplos de respuesta temporal ante pulso de duración 0.01 segundos y analiza también el diagrama de Bode para entender cuando NO debe utilizarse el concepto de dominancia.

[291: zercanc] Cancelación y ceros alejados: ejemplo y simulaciones de efecto en respuesta temporal sistema lineal

**  14:54
[YouTube ▶]

Este vídeo discute el efecto de los ceros (raíces del numerador) en el sistema lineal de función de transferencia $G(s) = 2(s/z + 1)/(s + 1)/(s + 2)$, que tiene un cero en ' $-z$ '. Se observa que cuando $z \rightarrow \infty$ entonces $G(s) \rightarrow 2/(s + 1)/(s + 2)$ y que cuando $z \rightarrow 1$ entonces $G(s) \rightarrow 2/(s + 2)$, y cuando $z \rightarrow 2$ entonces $G(s) \rightarrow 1/(s + 1)$. Se ilustra el concepto tanto con descomposición en fracciones simples y transformada de Laplace inversa, como con simulaciones de respuesta ante escalón (en la segunda mitad del vídeo).

Más detalle y ejemplos sobre el efecto de los ceros en la respuesta de los sistemas puede verse en los vídeos [142], [143], [144], [146].

9.2 Caso multivariable: planteamiento

[292: romot] Reducción de orden: motivación y aplicaciones

Materiales: [ModRed.pdf]

**  12:20
[YouTube ▶]

Este vídeo discute la motivación de por qué puedo tener modelos de trabajo iniciales de orden elevado y la conveniencia de su reducción.

En concreto, los modelos de elementos finitos (por ejemplo, los vídeos [78] y [278] que obtienen un modelo de un muelle de orden 80 que “mejora” al modelo $F = kx$ válido a baja frecuencia), o modelos con muchas leyes físicas (gran parte de las cuales manejan un porcentaje “despreciable” de la energía intercambiada entre los distintos elementos) resultan de un orden elevado (decenas o incluso cientos o miles de estados). Estos modelos son complicados de explicar y documentar a terceros, son lentos de simular, dan lugar a controladores “frágiles” (sensibles a error de modelado), y podrían tener problemas numéricos (stiff integration, errores de redondeo al manejar un gran número de elementos de matrices en descomposiciones, autovalores, etc.) afectando a la exactitud de los cálculos con los mismos.


La parte final del vídeo plantea criterios a considerar para resolver el problema: escalado, criterios de frecuencia, controlabilidad y observabilidad.

Nota: la aparición de los denominados “estados no controlables” y la reducción de orden para obtener una “realización mínima” es un problema relevante en las fases de modelado de algunos sistemas de “índice superior”. Se aconseja al lector revisar los ejemplos del vídeo [30] para comprender mejor esos aspectos que motivan también alguna de las aplicaciones de la reducción de orden.

A veces, puede plantearse la idea de no invertir esfuerzo en tener un modelo complejo que luego vaya a ser simplificado; podría ser aconsejable visualizar el vídeo [190] que discute las alternativas de modelado de un calentador tubular y orientaciones dependiendo de la aplicación para elegir complejidad del modelo.

[293: ro1] Reducción de orden de modelos lineales: parte 1, planteamiento general

Materiales: [ModRed.pdf]

***  04:54
[YouTube ▶]

Este vídeo presenta un marco general para la reducción de orden de sistemas lineales: suponiendo particionado el estado en un grupo x_1 a mantener y otro grupo x_2 a eliminar, presenta las ecuaciones resultantes de la eliminación / residualización.

Nota: Las siguientes secciones 9.3 y 9.4 presentan el detalle teórico de dos enfoques a la reducción de modelos, en el marco general descrito en el vídeo [ro1(04:54)]. El lector “con prisas” podría revisar únicamente la reducción modal (bastante sencilla teóricamente, vídeo [romo(11:24)]) y obviar por el momento la teoría más compleja de la reducción equilibrada (vídeo [robal(19:04)]).

Para este lector con poco tiempo, se recomendaría que revisara la ayuda de Matlab de los comandos `hsvd`, `balred`, `freqsep` y los ejemplos “demos” de reducción de orden que los utilizan, o, en el contexto de estos materiales, que pasara directamente al caso de estudio de la vibración longitudinal de un muelle en los vídeos [feredm(15:23)] y [feredm2(10:59)] donde se utilizan los comandos citados sin preocuparse mucho de la teoría subyacente, sino solamente de su significado y utilización en la resolución de problemas concretos.

9.3 Reducción modal


[294: romo] Reducción de orden de sistemas dinámicos: parte 2, reducción Modal

Materiales: [ModRed.pdf]

***  11:24
[YouTube ▶]

Este vídeo presenta la reducción modal: basado en los polos, eliminar modos rápidos o/y asociados a altas frecuencias. El método resulta sencillo en representación en función/matriz de transferencia. Este tipo de reducción es la que, por ejemplo, se usa en la parte final del caso de estudio sobre un transformador del vídeo [194], y en el ejemplo del vídeo [296].

9.4 Reducción equilibrada


[295: robal]	Reducción de Orden de sistemas dinámicos lineales: parte 3, reducción equilibrada (balanced)	*****  19:04
	Materiales: [ModRed.pdf]	[YouTube ►]

En este video se demuestra cómo se realiza la reducción equilibrada.

- se revisa el significado de los gramianos de controlabilidad y observabilidad.
- se presenta la realización equilibrada: cambio de variable que consigue que ambos gramianos sean iguales.
- se introducen los Hankel singular values (HSV), y se propone eliminar aquéllos pequeños.
- Se enuncia el teorema de cota de error (en norma infinito) menor del doble de la suma de los HSV eliminados.
- Se remite a otras técnicas de reducción de modelos con otros criterios/pesos frecuenciales en la Robust Control Toolbox de Matlab.

Nota: existen otros tipos de reducción de orden, motivados por diferentes problemas de control robusto (que el error de aproximación del modelo reducido sea pequeño según diferentes definiciones). Por ejemplo, dada una representación factorizada $P = M^{-1}N$ (vídeo [incf1(09:33)]) de un proceso, la reducción equilibrada de M y N tiene determinadas propiedades de interés (ver comando de Matlab Robust Control Toolbox `ncfmr`, y la discusión sobre el mismo en el vídeo [incf2(20:21)]).


9.5 Ejemplos de código

[296: roml]	Reducción de orden de modelos lineales: parte 4, ejemplo Matlab (control systems toolbox)	****  28:32
	Materiales: [Cód.: ctrbobsveyredord.mlx] [PDF]	[YouTube ►]

Sobre el ejemplo del video [275], se realizan las reducciones modal y equilibrada, y se comparan respuestas temporal y en frecuencia de los modelos originales (orden 4) y reducidos. Se comprueba la acotación del pico de respuesta en frecuencia (norma ∞) del error de modelado de la reducción basada en la realización equilibrada.

Ejemplo de aplicación a sistemas de parámetros distribuidos

Los sistemas de parámetros distribuidos discretizados por elementos finitos o diferencias finitas dan lugar a sistemas dinámicos de orden muy alto. Aplicar las técnicas de reducción de orden aquí vistas suele ser imprescindible para incorporar dichos modelos como subsistemas en desarrollos más complejos de modo que la simulación no sea muy costosa computacionalmente, o para diseñar controladores para este tipo de procesos.

[297: feredm]	Reducción de orden: ejemplo modos vibración longitudinal de un muelle (1)	***  15:23
	Materiales: [Cód.: reduccorderMuelle.mlx] [PDF]	[YouTube ►]

Este vídeo discute la reducción de orden de un muelle modelado por elementos finitos como un sistema lineal de orden 80 para poder reproducir sus modos de vibración longitudinales cuando tiene un extremo fijo y se le aplica una fuerza en el otro extremo. Los detalles del modelado se revisan en el vídeo [78], y su análisis de propiedades en tiempo, frecuencia, polos, ceros, etc. se discuten en el vídeo [278]; de todos modos los primeros dos minutos resumen rápidamente la idea básica y, si no se dispone de tiempo, puede comprenderse este vídeo sin haber visualizado esos otros materiales.

Primero, se presenta la reducción de orden basada en la realización equilibrada, utilizando los comandos `hsvd` y `balred`. [el uso de `hsvd` no es recomendado a partir de Matlab R2021a; la gráfica que ese comando hacía en versiones previas ahora la hace el propio `balred`].

Se compara también la respuesta con la de un modelo muelle-amortiguador $F_{muelle} = (K + f \cdot s)x_{extremo}$, esto es $x_{extremo} = \frac{1}{K+f \cdot s} F_{muelle}$, de primer orden. Se evalúa el error de modelado aditivo (sistema menos sistema reducido) y multiplicativo (error aditivo dividido por el módulo de la resp. frec. sistema), interpretable en términos porcentuales.

A partir del minuto [11:45] se aborda la reducción modal con el comando `freqsep`, con o sin preservación de la ganancia estática. En este caso, da un resultado muy parecido a la reducción equilibrada.

Nota: La reducción aquí detallada describe los modos de vibración del muelle “en solitario”. Cuando tiene una carga en el extremo cuya masa sea significativamente mayor que la del muelle, puede reducirse todavía más el sistema (y despreciar *todos* los modos de vibración resonantes del muelle), el detalle de esto se aborda en el vídeo [298] a continuación.

[298: feredm2]

Reducción de orden: ejemplo modos de vibración de muelle (2),
con carga en un extremo

Materiales: [[CÓD.:](#) reduccorderMuelle.mlx] [[PDF](#)]

10:59

[\[YouTube ►\]](#)

Este vídeo es continuación del vídeo [297], donde se obtenían modelos de orden 2 o 4 de un muelle cuyo modelo de elementos finitos era originalmente de orden 80.

En el nuevo vídeo ahora en consideración, se estudia el comportamiento de dicho muelle cuando tiene adosada una masa M_2 en uno de sus extremos y se aplica la fuerza F_2 a dicha masa.

El muelle modelado tenía una dinámica $p(s) = G_{muelle}(s)F_{muelle}(s)$, siendo p la posición del extremo del muelle, siendo G_{muelle} de orden 80.

Una masa que tiene a un lado el muelle y al otro lado se le aplica una fuerza F_2 tiene de ecuación $M_2 \frac{d^2 p}{dt^2} = F_2 - F_{muelle}$, siendo la posición de la masa la misma que el extremo del muelle si están unidos mecánicamente.

Entonces, a partir de lo de arriba, despejando $p(s)$ se tiene

$$p = \frac{1}{M_2 s^2 + G_{muelle}^{-1}(s)} F_2$$

El vídeo estudia las dinámicas resultantes al sustituir en G_{muelle} los modelos de orden 1 (muelle-amortiguador sin masa ni estructura interna), de orden 2 y 4 a partir de `balred`, y el modelo original de orden 80. Se comprueba, cuando la masa cargada es mucho mayor que la del muelle que todas las respuestas son prácticamente coincidentes y que un nuevo `balred` justifica utilizar un modelo completo de orden 2 con un sólo modo de vibración (esto es, básicamente el “masa-muelle-amortiguador” clásico “de toda la vida” que es el ejemplo típico en los primeros cursos de modelado de sistemas).

A partir del minuto [07:30] se analiza el caso más interesante de que la masa en el extremo del muelle sea “comparable” a la del propio muelle. En ese caso, ya se observa que los modos de vibración y la estructura interna del muelle ya no son despreciables: el modelo

de muelle-amortiguador sin masa (de primer orden) se equivoca bastante en la predicción de la frecuencia de vibración del conjunto, mientras que todos los modelos basados en el modelo de elementos finitos o sus aproximaciones de orden 2 o 4 predicen la misma primera resonancia a la frecuencia correcta.


Parte II

Control Monovariabile

Capítulo 10

Reguladores monovariables en tiempo continuo

10.1 Definiciones básicas de sistemas de control

[299: contrintu1]	Problemas de control en un proceso industrial: discusión y definiciones intuitivas Materiales: [ControlIntuitivoDefProblemaBreve.pdf]	*  17:50 [YouTube ▶]
-------------------	---	--


Este vídeo discute qué es el “control” en términos intuitivos, sin fórmulas: en un proceso industrial, hay tareas de *ingeniería de proceso* de control, monitorización/supervisión y de mantenimiento y actualización tecnológica de subsistemas.

El control consiste en mantener un punto de operación frente a errores de modelado o perturbaciones, en alcanzarlo (con suficiente rapidez) si no se está en él, y en estabilizar al sistema si no fuera estable. El detalle científico de estos problemas son el contenido usual de las primeras asignaturas de grado universitario de control automático.

Refinamientos del control serían seguimiento de trayectorias, secuenciación de operaciones (*automatización industrial*) y control en tareas de seguridad (*override*) si por alguna razón el proceso se acerca a límites tecnológicos peligrosos.

Algunos problemas se ilustran con una animación de un control de nivel de un depósito, similar al usado en el vídeo [5], que introduce conceptos básicos de señales y sistemas “sin fórmulas” y quizás deberías ver también.


El vídeo [300], continuación de este, describe algunas de las opciones tecnológicas en la industria para abordar problemas de control.

[300: contrintu2]	Problemas de control de procesos industriales: opciones tecnológicas para abordarlo Materiales: [ControlIntuitivoDefProblemaVideo2.pdf]	*  15:41 [YouTube ▶]
-------------------	---	--

Este vídeo, continuación del vídeo [299], discute qué opciones tecnológicas hay para abordar la ejecución de tareas asociadas al control:

- Un buen diseño del proceso para que se reduzcan las necesidades de control
- Mecanismos ingeniosos originados a lo largo de la historia
- Control manual (operador humano)
- Control automático con tecnología electrónica e informática

El tipo de reglas o estrategias de control concretas (PID, PLC, ...) no son objetivo de este vídeo, donde sólo discutimos, una vez definido qué es “controlar”, con qué herramientas/personas deberíamos hacerlo.

[301: fundcontrR] Fundamentos del Control Realimentado Instrumentacionycontrol.net **  10:21
[PLAY ►]


Los vídeos [299] y [300] introducen qué es el “control de procesos” y qué opciones tecnológicas existen en industria.

Este vídeo introduce los elementos básicos de un bucle realimentado (sensor, actuador, variable controlada o variable de proceso, y variable manipulada, controlador), sobre un ejemplo de control de temperatura.

► *Prealimentación (feedforward) y realimentación (feedback):*

[15:58]
[Enlace]

10.2 Control en Bucle Abierto

[302: tank1BAP] Control nivel tanque de líquido en bucle abierto: interpretación intuitiva, casos proporcional y no lineal **  20:30
Materiales: [CÓD.: ControllerOpenLoop.m] [PDF] [YouTube ►]

Este vídeo simula una animación de un control en bucle abierto (el más simple, sin dinámica) de nivel de un tanque de líquido. La variable manipulada es el caudal de entrada superior, la controlada es el nivel, y existe una entrada perturbación (incremento de válvula de salida). El código de simulación interactivo se basa en el que se discutió en el vídeo [40] y el ejemplo sirvió para motivar hacia los conceptos básicos de señales y sistemas en el vídeo [5].

La primera parte del vídeo discute un control $u = K_r \cdot r$ donde se ejerce una acción (incremental) proporcional a la referencia (también incremental). La constante K_r es la inversa de la ganancia estática del sistema, estimada mediante un ensayo escalón de pequeña amplitud alrededor de un punto de funcionamiento en el que se ha posicionado el sistema con control “manual”.

Se presenta el código y se comprueba un funcionamiento correcto para incrementos de nivel pequeños alrededor de $h = 1$, como era de esperar. No obstante, cuando los incrementos de nivel son elevados, entonces el cálculo lineal en bucle abierto no posiciona el nivel donde se desea, porque la no linealidad dependiente de \sqrt{h} hace que el caudal de salida lejos del punto de funcionamiento sea sobreestimado, con lo que el nivel que se alcanza es superior porque se introduce más caudal del necesario.

La parte final del vídeo (minuto [11:30] en adelante) discute la idea de control en bucle abierto no lineal donde $u = \kappa_0 \sqrt{ref}$, esto es, que el caudal de entrada sea igual al caudal de salida que equilibraría un nivel de referencia dado. Obviamente, esto necesita un modelo no lineal del proceso, que puede ser más difícil de obtener con la debida calidad en la práctica (el caso lineal se obtenía con un ensayo escalón experimental)... pero si se dispone de ese modelo no lineal, es tontería no usarlo: qué más da hacer una multiplicación que una raíz cuadrada si se hace por computador.

Las simulaciones comprueban que ese bucle abierto no lineal, en efecto, consigue sin error el punto de nivel deseado.

No obstante, como conclusión del vídeo, salen a la luz dos inconvenientes bastante graves del control en bucle abierto aquí propuesto:


- Es lento: la dinámica del proceso es lenta y se debería “sobreactuar” en los instantes iniciales de algún modo. Un enfoque intuitivo al cálculo de ese ‘boost’ inicial para acelerar la respuesta se aborda en el vídeo [152].
- Al no utilizar el “sensor” de nivel, el incremento de válvula de salida desvía muchísimo el nivel del tanque, y el control no “reacciona” porque no se mide nada.

Solucionar esos problemas requiere cambiar la dinámica y medir cosas en bucle cerrado, objetivo de otros vídeos. Por ejemplo, el vídeo [306] analiza desde un punto de vista intuitivo, sin fórmulas, el control proporcional en bucle cerrado de este mismo tanque de líquido.

10.3 Control Todo-Nada (On-Off)

[303: onoffintro]

El control realimentado ON-OFF (Todo-Nada)
Instrumentacionycontrol.net


**  03:58

[PLAY ►]

Este vídeo introduce las características básicas del control ON-OFF (todo-nada), discutiendo sus ventajas (sencillez) y sus inconvenientes (oscilaciones, imprecisión, pérdidas de energía, durabilidad de componentes). Un ejemplo de control de nivel se desarrolla en el vídeo [304].

[304: tank1OnOff]

Control ON/OFF de nivel de tanque de líquido:
análisis/animación del comportamiento
Materiales: [[CÓD.:](#) ControllerOnOff.m] [[PDF](#)]

*  14:54

[YouTube ►]

Este vídeo ilustra el control todo-nada de nivel de un tanque de líquido.

Primero, se discute un control “naive” donde no hay una “banda” de error. Se producen oscilaciones de muy alta frecuencia (infinita si fuera de primer orden, aquí hay un período de muestreo del control y dinámica adicional de válvula y por tanto la frecuencia de las oscilaciones es “finita” pero grande).


La solución práctica para evitar fallos por fatiga de elementos electromecánicos es incluir una banda de error de modo que la conmutación del control ON/OFF se haga si el error sale de dicha banda. Con ello la frecuencia de las conmutaciones se reduce, a cambio de aumentar la amplitud de las oscilaciones del error.

La parte final del vídeo discute el código Matlab que ejecuta este tipo de control (es sencillo, solamente dos o tres “if else end”).

Las ventajas e inconvenientes del control TODO-NADA (on/off) se discuten en el vídeo [305].

[305: onoffVI]

Control Todo-Nada (On/Off): resumen ventajas e inconvenientes
Materiales: [OnOffVentajasInconv.pdf]

*  05:14

[YouTube ►]

Este vídeo discute brevemente las ventajas e inconvenientes del control todo/nada (ON/OFF), ampliamente usado en termostatos, control de presión, control de nivel, etc. Un ejemplo de control de nivel de un tanque se desarrolla en profundidad en el vídeo [304].

Las ventajas son la simplicidad y el buen comportamiento en sistemas de primer orden para los que no se requiera una gran precisión.

Las desventajas son la durabilidad de elementos mecánicos sometidos a conmutación rápida, la poca precisión si la frecuencia de conmutación es baja, las interferencias que la conexión y desconexión de elementos de alta potencia puedan hacer a otros sistemas, y que sólo funciona teóricamente bien para procesos de primer orden.

10.4 Reguladores PID: Comprensión “intuitiva” de las acciones básicas de control

Aunque hay toda una teoría detrás, los técnicos de Formación Profesional y, bueno, cualquier ingeniero que haya tenido contacto con sistemas de control y su ajuste, entienden ‘sin fórmulas’ cómo funcionan más o menos los sistemas de control PID. Esta sección trata de discutir dicha comprensión intuitiva con el mínimo número de referencias a la teoría subyacente, que se detallará en futuras secciones de este documento.

Control PROPORCIONAL

[306: tank1P] Control proporcional de nivel de tanque de líquido: simulación
Matlab y explicación intuitiva, error de posición
Materiales: [CÓD.: ControllerPv1.m] [PDF]

** 18:27
[YouTube ▶]

Este vídeo simula el control en bucle cerrado del nivel de un tanque de líquido, cuyo código de simulación interactivo en bucle abierto se discutió en el vídeo [40] y sirvió para motivar hacia los conceptos básicos de señales y sistemas en el vídeo [5]. El control en bucle abierto se aborda en el vídeo [302]... aunque tiene bastantes inconvenientes prácticos comparado con el bucle cerrado.

Aquí se plantea un bucle de control, primero en diagrama de bloques (aunque el análisis teórico de las funciones de transferencia de bucle cerrado NO es objetivo de este vídeo), y luego sobre el ejemplo concreto del tanque, identificando variable controlada (nivel), variable manipulada (caudal de entrada), perturbación (válvula/caudal de salida), etc.

Se discute que el control proporcional puede implementarse con un mecanismo flotador-palanca, pero también puede hacer una multiplicación $u = K \cdot error$ un computador... y además de multiplicar, registrar datos, presentar interfaz de usuarios, permitir teleoperación remota, etc.

Primero se simula el control $u = k \cdot (r - y)$, comprobando que tiene un gran error de posición. Realmente es incorrecto, porque las fórmulas lineales manejan variables incrementales por lo que la simulación correcta es $u = k \cdot (r - y) + u_{PF}$, esto es añadiendo un “offset” a la acción de control igual al valor de la entrada en el punto de operación nominal. Esta simulación se analiza, por tanto, pero también tiene error de posición si nos alejamos del punto de operación tomado como “cero” en incrementales, o si las perturbaciones de caudal de salida actúan.

La razón básica del error de posición es que el control proporcional “desconecta” el control al llegar al valor de referencia y, por tanto, ese nunca va a ser un punto de equilibrio (salvo que el “offset” al que referíamos en el anterior párrafo coincida con el punto de operación de entrada en la citada referencia... lo que no ocurrirá si dicha referencia cambia, obviamente). Ello requiere soluciones (acción integral, preprocesado de referencia, etc.) que no son objetivo de este vídeo.

*Los materiales incluyen sólo el código del controlador.

[307: Pintu] Control Proporcional sistema 1er orden estable, prueba y error
(ajuste empírico de PIDs)
Materiales: [CÓD.: PIDcontrolAPP1.0.zip]

** 18:59
[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo discute el control proporcional en un sistema de primer orden y su relación con control todo-nada. El proceso en concreto será $2.5/(2s + 1)$.

Primero, recuerda lo que es el comportamiento de un sistema de 1er orden estable (transitorio exponencial entre dos puntos de equilibrio, efecto instantáneo entre variable manipulada y velocidad de la salida)... nivel, presión, concentración son, en primera aproximación, ejemplos de esta dinámica.

Luego, recuerda el diagrama de bloques de un bucle cerrado usual, pero en sintonía por prueba y error no son necesarias las ecuaciones de bucle cerrado. En el bucle que se simulará se añade ruido de medida, saturación y perturbaciones a la entrada (ruido de proceso), y se representan en dicho diagrama de bloques.

A continuación, presenta la estructura de una aplicación de simulación de controladores que será utilizada en este vídeo y en sucesivos.

Se comienza probando reguladores proporcionales de ganancia creciente. Se observa que existe un error residual (error de posición) en régimen permanente que decrece conforme crece la ganancia; los transitorios son asimismo cada vez más rápidos.

La simulación presenta la limitación de rapidez causada por saturación y la amplificación de ruido de medida. Discute que cuando la ganancia es muy alta, converge a una especie de control ‘todo-nada’. Se discute la relación entre control proporcional y ON-OFF (o bi-nivel en caso genérico).


La parte final del vídeo [14:00] discute la corrección del error ante cambios de referencia mediante un cálculo correcto del ‘punto de operación de la entrada’ (denominado ‘manipulated offset’ en alguna documentación de controladores). Esto también se denomina control proporcional con ‘dos grados de libertad’... Pero NO sirve para cancelar el efecto de error debido a perturbaciones externas no medibles, porque si la referencia no se incrementa, no se puede hacer ningún cálculo o corrección. Para cancelar el error ante perturbaciones no medibles es necesario usar acción integral, pero eso no es objetivo de este vídeo.

Acciones Integral y Derivada

[308: carPIDMIT]

Control PID, comprensión intuitiva: ejemplo en conducción de vehículo

AerospaceControlsLab (MIT)

**  04:40

[PLAY ▶]


Este vídeo, en inglés, de un laboratorio del MIT, ilustra sin fórmulas las ideas principales y el tipo de comportamiento “intuitivamente esperable” que está asociado a las acciones de control proporcional, derivada e integral basándose en un ejemplo de seguimiento de trayectoria de un vehículo autónomo.

► *Acciones básicas de control proporcional, integral, derivativa:* [16:33]
[Enlace]

[309: tank1PIanim]

Control proporcional-integral (PI) de nivel tanque de líquido: animación, interpretación intuitiva

Materiales: [[CÓD.:](#) ControllerPI.m] [[PDF](#)]

**  16:57

[YouTube ▶]

Este vídeo discute de forma “intuitiva”, con pocas fórmulas, qué es la acción integral: una corrección supuestamente lenta de la acción de control si se acumula error. La acumulación es un integrador o, en código $u_i = u_i + K_I \cdot \text{error} \cdot T$ aproximando por Euler, siendo T el período de muestreo.

El vídeo simula la animación del control de nivel de un tanque que ya se había explorado con control proporcional en el vídeo [306]. Con acción integral, se aprecia una sobreoscilación (quizás el valor de K_I sea excesivo para seguimiento de referencia, pero los

detalles de diseño de K_I no son nuestro objetivo en este momento), pero el error se hace cero en régimen permanente, como la teoría predice.

En efecto, la discusión “teórica” sobre este mismo problema de control se aborda en los vídeos [329] y [330], basándose en modelos linealizados y funciones de transferencia de bucle cerrado en el dominio de Laplace. Pero la comprensión “intuitiva” del funcionamiento de un PI no requiere dominar en profundidad esas manipulaciones teóricas.

La parte final del vídeo insiste en el fenómeno del “wind-up” donde la u_I crece sin límite en caso de saturación sin poder alcanzar la referencia pedida... Si la saturación es transitoria, también crece u_I más de lo que la teoría lineal predice y aumenta la sobreoscilación. La solución de ese problema no es objetivo de este vídeo, que sólo intenta ilustrar el efecto de que incluir acción integral hace el error cero (si el bucle continua siendo estable, claro).

[310: PIintu1]

Ajuste empírico/intuitivo control Proporcional Integral: proceso
dominantemente 1er orden (PI académico)

** 12:36

Materiales:

[CÓD.: PIDcontrolAPP1.0.zip]

[YouTube ►]

En este vídeo se desarrolla el ajuste de un controlador PI por ensayo y error para un proceso de primer orden. Bueno, en realidad el proceso simulado subyacente es de segundo orden $G(s) = 1.8/(2s + 1)/(0.2s + 1)$, para ilustrar que las heurísticas de primer orden (polo dominante) se aplican bastante bien, a menos que queramos acelerar nuestro sistema de lazo cerrado ‘mucho’: el tiempo de establecimiento en lazo abierto es de 6 segundos; entonces, si buscamos un tiempo de establecimiento en lazo cerrado de alrededor de 1.5 segundos, no hay nada que la constante de tiempo rápida en 0.2 haga, por lo que podemos operar ‘como si’ estuviéramos en un proceso de primer orden.

Por lo tanto, comenzamos por ajustar un controlador proporcional como hicimos en el vídeo [307]. Como hay un error de posición en estado estable (y no podemos aumentar la ganancia porque aparecen oscilaciones), dicho error se corregirá agregando acción integral, proponiendo así un controlador PI.

La acción integral mejora tanto el seguimiento de referencias como el rechazo de perturbaciones de entrada, hasta que alcanzamos un valor de ganancia integral que sigue mejorando el rendimiento del rechazo de perturbaciones pero comienza a empeorar (sobreoscilación) la respuesta al escalón de referencia. Por lo tanto, existe un compromiso entre la velocidad con la que se rechazan las perturbaciones y el sobrepaso en la respuesta al escalón. La saturación también tiene un efecto perjudicial en la respuesta del sistema. En el siguiente vídeo [311] se analizarán algunos ajustes para superar estos problemas (dos grados de libertad y antiwindup).

Aunque teóricamente los conceptos de ‘anti-windup’ y ‘control con 2 grados de libertad’ son abordados en asignaturas más avanzadas, ello no impide entender su utilidad práctica ‘sin fórmulas’ viendo su efecto sobre simulaciones, del mismo modo que alguien puede utilizar un teléfono móvil sin comprender las *ecuaciones de Maxwell o Dirac del electromagnetismo*. Por tanto, resulta de interés que, opcionalmente, visualices el siguiente vídeo:

[311: PIintu2]

Proporcional Integral intuitivo: proceso dominantemente 1er
orden (PI 2GL+antiwindup)

*** 12:17

Materiales:

[CÓD.: PIDcontrolAPP1.0.zip]

[YouTube ►]

Este vídeo continúa el estudio de caso del vídeo [310], donde se diseñó el control proporcional-integral de un proceso (dominantemente) de primer orden a través de simple intuición y el ensayo y error.

En este vídeo, se explican y verifican intuitivamente las mejoras que se obtienen al usar antiwindup cuando se está sujeto a saturación (no se presenta ninguna teoría sobre el

método de cálculo de dicho antiwindup); además, se prueba un PI como $u(s) = br(s) - y(s) + (1/s) \cdot (r(s) - y(s))$ de dos grados de libertad para disminuir la sobreoscilación en la respuesta ante referencia mientras se mantiene una corrección rápida de acción integral para un buen rendimiento en el rechazo de perturbaciones de entrada.

En inglés, por el momento...

[312: dintpid1motEN] Double-integrator and its control (1): motivation
17:54



[YouTube ▶]

This video discusses the relevance in engineering practice of studying the control of a ‘double integrator’ system, transfer function $1/s^2$.

Indeed, the motion of a mass $M \frac{d^2p}{dt^2} = F$ has such dynamics, if there is no friction... and, well, in technological applications of motion control one should try to make such friction as small as possible, at least in principle. Angular motion also has double integrator dynamics, changing masses for moments of inertia and forces for torques, obviously.

The transfer function $1/s^2$ is an ‘abstraction’: in many technological solutions, directly manipulating a force or torque is not possible, and an internal electronic system for ‘torque’ or ‘pressure’ control will be required, for example. This is called *cascade control* and is VERY frequently used to simplify and divide the complexity of a practical control problem into different ‘hierarchical levels’ where each level assumes that the lower slave controllers behave correctly so that they follow their references for force, torque, pressure, etc.

As a final example, in lateral position control of a drone, the non-vertical component of the force is approximately the weight times the sine of the roll angle; lateral attitude control of a drone may assume an A/s^2 model if there is internally a control system ‘much faster’ than the attitude control system that achieves a given pitch/roll angle using the appropriate accelerometers and gyroscopes.

All of this video is about the concept of a double integrator (low-friction motion control) in the abstract sense, although the ‘real technology’ of ‘motion control’ applications requires many more implementation details that are not the goal of a first introductory course in control theory, obviously.

In applications with high friction, gearboxes, etc., it is sometimes more appropriate to think of my manipulated variable as the ‘velocity of motion’, that is, to control position thinking of the transfer function $1/s$ (simple integrator); this type of abstraction is called ‘kinematic control’, in materials on robotics, for example. This is not considered in the present video.

[313: dintpid2tunEN] Double integrator and its control: trial and error controller tuning [1: PD]



18:37

Materiales: [Cód.: PIDcontrolAPP1.0.zip]

[YouTube ▶]

This video discusses the tuning of a PD controller for a ‘double integrator’ type process (mass motion with small, negligible friction). The motivation for studying this process has been addressed in the video [??], since motion control is an important problem in practical applications.


Here, the PID control problem of this process is starting to be discussed. This will be done through purely ‘intuitive’ considerations, common sense. The theory will be discussed in the video [331] and following. Actually, for brevity, the present video will discuss only the PD proportional-derivative tuning, and integral action plus final discussion will be left for video [314], a continuation of this one.

The basic intuitive consideration is that a proportional control is ‘analogous’ to a spring on the moving mass. Therefore, as can be seen in simulation, without friction, the response will show sustained oscillations: the P control does not succeed in stabilizing.

To stabilize, ‘damping’, friction... is needed and that is the analogue of the derivative action: a ‘braking’ proportional to the speed. Braking can be proportional to speed or proportional to the ‘error speed’, which gives rise to two possible implementations of the derivative action, parameterized by a certain coefficient ‘c’, which are tested in simulation. In the end, the important thing is that the derivative action does stabilize. Note that the ‘ideal’ derivative action is not feasible: the high-frequency noise is infinitely amplified and, therefore, a noise filter is necessary in all practical applications, as shown in simulations here.

A summary of the ideas concludes the video. Final value justification in intuitive terms, disturbance rejection and integral action are left for a sequel video.

[314: dintpid2tunBEN]
11:25

Double integrator and its control: trial and error PID tuning [2, *** 
PID; 3., advanced tweaks]

Materials: [[CÓD.:](#) PIDcontrolAPP1.0.zip]

[YouTube ►]

This video constitutes ‘Part 2’ of the trial-and-error heuristic tuning of a PID controller for a double-integrator (motion control) plant; PD control was discussed in video [313], and in here, we’ll discuss integral action for zero offset in disturbance rejection, antiwindup, and a final summary, conclusions and remarks.

First, it is intuitively justified that the final value in response to reference changes does achieve a stationary position error equal to zero (justified by the fact that when one is in place and at zero speed, then zero force is required to stay there, which is exactly what a PD regulator does). However, in response to disturbances there is a constant position error... the explanation is by analogy with a tilted platform: then at the reference point zero force is not the required value to achieve equilibrium, so PD has an offset when equilibrium is reached. This error is corrected by the integral action.

The final part of the video discusses that integral action to optimize the response to disturbances (unmodeled input forces) worsens the response to reference and that, to avoid overshoot due to error accumulation, the proportional part can be ‘fooled’ by lowering a certain parameter ‘b’.... Saturation also requires antiwindup (to compensate for excessive integral accumulation in the initial phases of the transient).

With all this, a PID regulator with derivative filter and with ‘2 degrees of freedom’ and ‘antiwindup’ modifications is finally designed (in a completely ‘math-free’ way) that achieves a satisfactory response to step, ramp and unmodeled step force changes (input disturbances).

‘Intuitive’ tuning strategies are extremely important in practice, because, in simple processes, it could be a quick option to solve a problem by a technician not specialized in control theory.

10.5 Demostraciones PID en equipos reales (de terceros)

Algunos vídeos de terceros:

[315: pitchpid] Pitch control using a PID

DIYmaker1 (YouTube)

00:25
[PLAY ►]

Un short con el control de la inclinación de un ventilador

[316: piddemo1]	Hardware demo of a digital PID controller Gregory L. Holst (YouTube)	02:57
		[PLAY ►]


Control de posición angular de un motor eléctrico, efecto PID y antiwindup

[317: piddemo2]	Position control, PID tuning Horizon 4 Electronics (YouTube)	01:20
		[PLAY ►]

Corto, efecto de distintas acciones PID en la posición angular.

10.6 Control PID: metodologías teóricas de diseño y ejemplos Matlab

Control Proporcional sistemas de 1er orden, teoría


[318: Pteo1]	Control Proporcional sistema 1er orden (teoría, 1): seguimiento de referencia Materiales: [CÓD.: JustifTheoryP1stOrderESP.mlx] [PDF]	**  19:58
		[YouTube ►]

Este vídeo analiza la teoría que justifica el comportamiento en lazo cerrado de sistemas de primer orden bajo una ley de control proporcional (P). El estudio se divide en dos vídeos: éste (enunciado del problema y respuesta ante cambios de referencia) y un segundo [319], que trata sobre el rechazo de perturbaciones a la entrada y la amplificación de alta frecuencia del ruido de medición.

En este primer vídeo, presentamos el diagrama de bloques y las funciones de transferencia en lazo cerrado de la variable controlada, la variable manipulada y el error del lazo de control como salidas con respecto a las entradas de referencia, perturbación de entrada y ruido de medición.

A continuación, se calcula la respuesta de referencia a salida controlada $y(s) = GK/(1 + GK) \cdot r(s)$ para un ejemplo particular con $G(s) = 2/(2s + 1)$, y se analiza su estabilidad, tiempo de establecimiento y valor final (error de posición).

El ejemplo es el mismo que utilizamos en el ajuste del controlador por "prueba y error" en el vídeo [307], por lo que podemos confrontar las reglas empíricas basadas en pura intuición de dicho vídeo con el respaldo teórico que se muestra aquí.

[319: Pteo2]	Control Proporcional sistema 1er orden (teoría, 2): rechazo perturbaciones y ruido de medida Materiales: [CÓD.: JustifTheoryP1stOrderESP.mlx] [PDF]	***  17:55
		[YouTube ►]

Este vídeo continúa el análisis, basado en el proceso ejemplo $G(s) = 2/(2s + 1)$ del comportamiento del control proporcional en procesos de primer orden estables. En el vídeo anterior [318] se habló de las prestaciones estáticas y dinámicas del seguimiento de referencias (análisis de $y = GK/(1 + GK) \cdot r$) y, en este vídeo, se detalla el rechazo de perturbaciones de entrada (análisis de $y = G/(1 + GK) \cdot d_u$), tanto en dinámica (tiempo de establecimiento) como en desviación en régimen permanente (valor final, ganancia de CC); por último, se habla de la amplificación del ruido de medida en la variable manipulada (análisis de respuesta en frecuencia de $u = K/(1 + GK) \cdot n$), sobre todo a altas frecuencias.

Realmente, la amplificación del ruido de medida suele ser un problema práctico en control proporcional-derivativo más que en el simplemente proporcional, pero aquí hacemos una primera introducción al problema y su naturaleza fundamentalmente frecuencial.

Caso estudio, control proporcional biorreactor inestable

[320: bio1cP] Sistema inestable 1er orden: control proporcional de bioreactor

**  21:27**Materiales:** [[CÓD.:](#) PIDcontrolAPP1.0.zip]

[YouTube ►]

Este vídeo describe las características básicas del modelo de un bioreactor de primer orden inestable (detalle en vídeo [77]): polos de bucle abierto, puntos de equilibrio. Se aborda tanto en ‘representación interna’ (ecuación diferencial) $\frac{dx}{dt} = 0.4x + u$ como en ‘función de transferencia’ $x(s) = \frac{1}{s-0.4}u(s)$.

Como el equilibrio es inestable, para poder ser utilizado manteniendo una biomasa constante, el sistema debe ser estabilizado. Para ello, se analizará el comportamiento de un control proporcional $u(x) = K_c(r - x)$. Se comprueba que los polos de bucle cerrado son $0.4 - K_c$, por lo que el bucle cerrado es estable con $K_c > 4$, y que el control proporcional tiene un error de posición dado que el valor final de la salida es $x_{eq} = \frac{K_c}{K_c - 0.4}r$, que para $K_c > 0.4$ es mayor de r (sobrepasamiento en régimen estacionario).

La corrección de ese error y la respuesta ante perturbaciones serán analizadas en futuros materiales.

[321: bio1tgl] Sistema inestable 1er orden (biorreactor): control proporcional 2GL, perturbaciones

***  17:50**Materiales:** [[CÓD.:](#) PIDcontrolAPP1.0.zip]

[YouTube ►]

Este vídeo discute los conceptos de control proporcional con 2 grados de libertad y analiza la respuesta del control proporcional ante perturbaciones escalón a la entrada del proceso. El estudio se enmarca dentro de un ejemplo de biorreactor inestable de primer orden, caso de estudio que comenzó en el vídeo [77] y cuyo control proporcional ‘básico’ se abordó en el vídeo [320].

Como se vio en dicho vídeo, el control proporcional $u = K_{error}$ tiene error de posición en régimen permanente ante cambios escalón de referencia. Aquí se revisan rápidamente las conclusiones del vídeo [320] y se calcula teóricamente dicho error de posición. Después, se argumenta ‘intuitivamente’ que una forma de solventar el problema (ante referencia) es hacer creer, al software de control, mediante un ‘engaño’, que la referencia es diferente a la que introdujo el operador, con una fórmula $u = K_c \cdot (br - y)$, bueno, en este caso $y = x$ es la biomasa. Tras la argumentación intuitiva, se detalla el desarrollo teórico para entender que, en efecto, eso solventaría el error de posición.

La base de la teoría es que el control $u = K \cdot e$ realmente está ‘mal implementado’, porque el punto de operación deseado debería ser ‘cero’ en control lineal: la forma correcta debería ser $\Delta u = -K_c \cdot \Delta y$ en coordenadas ‘incrementales’, con lo que el control proporcional ‘correctamente formulado’ debería ser $u = \Delta u + u_{eq} = K_c \cdot (r - y) + u_{eq}$ siendo este término u_{eq} omitido en muchas de las implementaciones ‘naïve’ del control proporcional, sólo para ser añadido posteriormente en refinamientos, pero denominado como ‘input offset’, ‘input trimming’ o ‘control con 2 grados de libertad’. Esto soluciona el error estacionario ante seguimiento de referencias constantes, pero no soluciona las desviaciones debidas a perturbaciones. El objetivo del vídeo es el detalle de todo esto sobre el ejemplo del biorreactor.

Control Proporcional-Integral del biorreactor:

[322: bioPI1] Sistema inestable 1er orden (biorreactor): control proporcional-integral (PI)

**  22:31**Materiales:** [[CÓD.:](#) PIDcontrolAPP1.0.zip]


[YouTube ►]

Este vídeo presenta la teoría subyacente a la dinámica en bucle cerrado de un modelo inestable de primer orden $1/(s-0.4)$, inspirado en un biorreactor, con control proporcional-integral.

Se discuten las siguientes cuestiones:

- La sintonía PI por ‘ensayo y error’ heurística
- La obtención de las ecuaciones del error en bucle cerrado
- Las condiciones de estabilidad
- Comprobación de que el error estacionario es cero (si el bucle es estable)
- Análisis de prestaciones (tiempo de establecimiento) de un regulador dado
- Diseño de un regulador para tiempo de establecimiento dado requerido

[323: bioPIrloc] Sistema inestable 1er orden: control proporcional-integral lugar de las raíces (biorreactor)

***  20:02

Materiales: [[CÓD.:](#) PIDcontrolAPP1.0.zip]


[YouTube ►]

Este vídeo plantea el ‘lugar de las raíces’ del modelo de primer orden $G(s) = 1/(s - 0.4)$ de un biorreactor (inestable), cuando es controlado con un regulador proporcional-integral (PI) de tipo $K(s) = K_c \cdot (1 + K_i/s)$, esto es $K_c \cdot (s + K_i)/s$, comparando en simulación con las conclusiones sobre cómo cambian las raíces de $1 + G(s)K(s) = 0$ cuando varía K_c (en la primera parte del vídeo) y cuando varía K_i en la parte final del vídeo.

Realmente, el bucle cerrado es de orden 2 y, por tanto, no es necesario recurrir a la metodología de lugar de las raíces dado que dichas raíces pueden ser obtenidas analíticamente con la fórmula ‘de toda la vida’ de las raíces de una ecuación cuadrática; de hecho, eso es lo que se ha hecho en el vídeo [322], que deberías haber visionado previo a éste. El uso del lugar de las raíces aquí es simplemente como ejemplo motivador para hacer la conexión con desarrollos anteriores; la utilidad de la metodología es mayor en sistemas de tercer orden o superior, en los que los polos de bucle cerrado no pueden calcularse (o no resulta fácil) con una expresión analítica sencilla.

Más ejemplos orden 1

[324: ord1ejP] Control proporcional de sistema lineal de primer orden: ejercicio resuelto (Matlab)


**  11:35

Materiales: [[CÓD.:](#) EjerciciosPyPIord1.mlx] [[PDF](#)]

[YouTube ►]

Este vídeo describe un sencillo ejercicio académico sobre qué especificaciones son factibles con control proporcional en un proceso lineal estable de primer orden. Básicamente, cualquier error de posición no cero (detalles vídeo [356]) y tiempo de establecimiento son factibles con control proporcional, subir ganancia hace bucle más rápido y con menor error (“en teoría”... las ganancias excesivamente elevadas dan “problemas prácticos” de saturación, amplificación de ruido y poca tolerancia a error de modelado, que no son objetivo de este material introductorio).

[325: ord1ejPI] Control Proporcional-Integral de sistema lineal de primer orden: ejercicio resuelto (Matlab)

**  14:12

Materiales: [[CÓD.:](#) EjerciciosPyPIord1.mlx] [[PDF](#)]

[YouTube ►]


Este video aborda un diseño de control para el mismo proceso de primer orden $G(s) = 3/(s+1)$ que el discutido en el vídeo [324]. La diferencia es que aquí se pide un error de posición nulo, por lo que es necesaria acción integral. Se discute cómo calcular un control PI para error de posición cero y tiempo de establecimiento de 0.5 segundos, por dos métodos:

- Con regulador genérico $K(s) = K_p + K_i/s$, asignando DOS polos
- Con regulador $K(s) = K_p \cdot (s+1)/s$ donde se cancela el polo de la planta de modo que el problema puede resolverse rápidamente asignando UN polo.

Caso de estudio control nivel tanque de líquido

[326: tank1ModyBA]
15:14

Modelado teórico, análisis y control en bucle abierto de nivel de un tanque de líquido

** 

Materiales: [[CÓD.:](#) ClosedLoopTeoricoP.mlx] [[PDF](#)]

[\[YouTube ▶\]](#)

Este vídeo comienza el análisis “teórico” del comportamiento en bucle abierto y bajo control de un tanque de líquido, cuya simulación “intuitiva” sirvió para introducir conceptos de señales y sistemas en el vídeo [5], y para el cual se diseñaron controladores sencillos proporcionales en bucle abierto en el vídeo [302] y en bucle cerrado en el vídeo [306]. Estos vídeos discutían el problema desde un punto de vista “intuitivo”, sin fórmulas en el dominio de Laplace, y se aconseja su visualización.

El objetivo de este vídeo es comenzar con los desarrollos teóricos que permitan justificar los comportamientos observados y explicados por “sentido común” en los citados vídeos.

En concreto, en este vídeo se modela el tanque como un sistema de primer orden $\frac{dh}{dt} = \frac{1}{6}(u - \kappa\sqrt{h})$, siendo u y κ variables de entrada, y h variable de estado. Se escoge un punto de funcionamiento, se linealiza, se obtiene una representación interna normalizada y una matriz de transferencia.


Se separa el comportamiento en $y = G(s) \cdot u + G_d(s) \cdot d$ dando rol de variable manipulada (acción de control) a la entrada u (caudal de entrada superior) y dando el rol de perturbación a la entrada d (incremento de válvula de salida, esto es, incremento de κ). Estos roles serán importantes en “control”, pero no lo son para “simulación” donde son simplemente “entradas” al sistema.

La parte final del vídeo analiza las propiedades en bucle abierto (ganancia –vídeo [??]– y tiempo de establecimiento). Como corolario de dicho análisis, se plantea un control en bucle abierto como $u = \frac{1}{G(0)}r$ para que se alcance la referencia r de nivel en régimen permanente cuando d sea cero. Obviamente, este análisis teórico es análogo al análisis “intuitivo” del control en bucle abierto lineal propuesto en el vídeo [302] donde se simula y se presentan los conceptos basados en una animación interactiva.

El uso de este modelo para el cálculo teórico de funciones de transferencia de *bucle cerrado* se aborda en el vídeo [327]. El mismo cálculo en el caso de control proporcional-integral y dinámica de actuador se discute en el vídeo [329].

[327: tankCLEq]

Ecuaciones en bucle cerrado: seguimiento de referencia y rechazo de perturbaciones. Ejemplo Matlab control nivel líquido.

**  18:51

Materiales: [[CÓD.:](#) ClosedLoopTeoricoP.mlx] [[PDF](#)]

[\[YouTube ▶\]](#)

Este vídeo calcula las ecuaciones de bucle cerrado $e = \frac{1}{1+GK}r - \frac{G_d}{1+GK}d$, así como las de la acción de control u y de la salida controlada y , a partir de las ecuaciones de bucle abierto $y = Gu + G_d d$, y el controlador $u = K \cdot (r - y)$.

Aunque los resultados son de aplicación genérica, la motivación y el cálculo concreto usando Matlab se han basado alrededor de un problema de control de nivel de un tanque de líquido que se aborda en otros vídeos complementarios a éste, como [306] que aborda el control proporcional desde un punto de vista “intuitivo”, o el vídeo [326] que deriva el modelo en bucle abierto que se usará en la parte final del vídeo para presentar un ejemplo concreto de las ecuaciones de bucle cerrado objeto de estudio, con control proporcional.

También en la parte final del vídeo se ilustra el uso del comando `connect` para interconectar “por nombre” elementos de un bucle de control, obteniendo un resultado idéntico a las derivaciones algebraicas de la teoría.

El análisis de las prestaciones en bucle cerrado del tanque de líquido con control proporcional se discute en el vídeo [328], continuación de éste; aquí únicamente el objetivo es obtener las funciones de transferencia de bucle cerrado, no interpretarlas para decidir si el control funciona bien o no.

El vídeo [329] discute las ecuaciones en bucle cerrado con `connect` en el caso de dinámica de actuador, también considerando controladores con acción integral.

[328: tank1CLan] **Análisis linealizado de respuesta en bucle cerrado de control de nivel con control proporcional**

Materiales: [[Cód.:](#) ClosedLoopTeoricoP.mlx] [[PDF](#)]

*** 18:49

[YouTube ▶]

Este vídeo, continuación del [327], analiza el significado de las funciones de transferencia de bucle cerrado que se calcularon en el vídeo referido para un problema de control de nivel de un tanque.

Se analiza el tiempo de establecimiento, la ganancia estática (errores de bucle cerrado), comprensión del comportamiento ante cambio de referencia y cambio de perturbación, el concepto de “esfuerzo de control”, y en la parte final se compara el control en bucle abierto (sin sensor) con el control en bucle cerrado proporcional. El bucle cerrado es más rápido y atenúa el efecto de las perturbaciones casi en un factor de 10... pero tiene un error de posición residual que, para eliminarlo, requiere complicar un poco más el controlador, añadiendo “acción integral”, objeto de otros materiales.

[329: tank1PI1teo] **Ecuaciones en bucle cerrado de control de nivel con control proporcional-Integral y dinámica de actuador**

Materiales: [[Cód.:](#) tankClosedLoopTeoricoPI.mlx] [[PDF](#)]

13:29 [YouTube ▶]

Este vídeo aborda el cálculo del modelo en bucle cerrado de un tanque de líquido sujeto a cambios de referencia de nivel y perturbaciones, con controladores de tipo proporcional+integral (PI).

El vídeo es paralelo al del vídeo [327], donde se obtenían las ecuaciones en bucle cerrado con control proporcional. Aquí, por diferenciarlo, también se añade “dinámica de actuador”, esto es, se supone que la orden del controlador pasa por una bomba o válvula que tardan un cierto tiempo en acelerar o abrirse para dar el caudal demandado. En concreto, el actuador tiene de función de transferencia $1/(0.25s + 1)$.

Realmente, la dinámica del actuador es dominada en un factor de 50 en el caso en bucle abierto (polo en -4 del actuador versus polo en -0.075 del llenado). En bucle cerrado, será despreciable si el polo dominante de bucle cerrado está en -0.4 o más lento, si no, podría tener efectos a tener en cuenta.

Las ecuaciones en bucle cerrado son obtenidas con el comando `connect`, que conecta los distintos elementos “por nombre”. Se calculan cuatro bucles cerrados:

1. Control Proporcional $K = 3$.
2. Tres controladores integrales $K_p + K_i/s$ con $K_p = 3$ y valores crecientes de K_i .

El análisis detallado de las propiedades en bucle cerrado resultantes se abordará en el vídeo [330], continuación de éste.

[330: tank1PI2step] Control de nivel con control proporcional-Integral y dinámica de actuador: análisis comparativo de opciones de ganancia integral

18:59

Materiales: [Cód.: tankClosedLoopTeoricoPI.mlx] [PDF]

[YouTube ▶]

Este vídeo continua el vídeo [329], donde se discutió el concepto de dinámica de actuador, el controlador PI, y la obtención con connect de los modelos en bucle cerrado en un problema de control de nivel de un tanque cuyo análisis teórico comenzó con el vídeo [326] (modelado).

En los tres primeros minutos se revisa el primer video antes citado, y en este vídeo se hace especial énfasis en el análisis de la respuesta ante escalón de referencia y ante escalón de perturbación (incremento válvula de salida), tanto de la variable de proceso controlada (y) como de la manipulada (u , esfuerzo de control).

El objetivo es comparar el control P con tres posibles PI, con diferentes ganancias integrales, a saber: “pequeña” (una corrección lenta del error), “mediana” (cancelación del polo dominante del proceso) y “grande” (interfiere con la dinámica natural del proceso y aparecen oscilaciones).

Se analiza valor final y tiempo de establecimiento, tanto “gráficamente” de forma “intuitiva” para ver si la cosa funciona bien como “analíticamente” calculando matriz de ganancia y polos de bucle cerrado (esta segunda opción sería la única posible en un examen de “lápiz y papel” sin acceso a Matlab, claro).

Caso de estudio PID ‘doble integrador’ (control de movimiento)

En elaboración/edición, próxima publicación en castellano.

Versión en inglés:

[331: dintteostEN] Double integrator, PD control: stability

20:59

Materiales: [Cód.: DINTJustifTheoryPandPI.mlx] [PDF]

[YouTube ▶]

This video is the first in a series to present the theory that justifies the behaviour we had found by ‘trial and error’ when tuning a PID controller for a ‘double integrator’ process $2/s^2$, see video [313] for such hand-tuning ideas.

In this first video on the underlying theory, the generic closed-loop equations $e(s) = \frac{1}{1+GK}r(s) - \frac{G}{1+GK}d_u(s)$ are obtained, and $G = 2/s^2$ is substituted; the controller is also substituted in there, several versions actually, as follows.

First, a proportional controller is tested, checking that the loop CANNOT be stabilized, having sustained oscillations (i.e., they do not decay to zero) with $K_p > 0$.


Then, a PD controller is tested checking that, if both proportional and derivative constants are positive, then it CAN be stabilized. With ‘little’ derivative action there are oscillations, but if the intensity of the derivative action is increased, then there are a pair of real poles (overdamped response).

The analysis continues in the video [332], where the steady-state error is analyzed with PD and PID controls. The design of controllers by pole assignment will be discussed in later videos of the case study.


[332: dintteoerrEN]

13:51

Double integrator, PD control: position and velocity errors
(setpoint tracking and disturbance rejection)

*** 

Materiales: [[CÓD.:](#) DINTJustifTheoryPandPI.mlx] [[PDF](#)]

[YouTube 

This video is a continuation of the video [331]; in that video the closed loop poles of a double integrator with PD control were analyzed (it was stable, oscillatory or not depending on the amount of derivative action); in the present video, the error in steady state is analyzed.

Applying the final value theorem with a control law $K_P e + K_D \frac{de}{dt}$ and $G(s) = 2/s^2$, the following situations are analyzed:

- Error with step (e_{pos}) and ramp (e_{vel}) setpoint changes. Both are zero with the control $K_P + K_D s$.
- Error with input disturbance step (non-zero final deviation) and ramp (infinite final deviation) increments.


The results are verified in simulation.


Note: if the derivative action were implemented as $K_D \frac{dy}{dt}$ (output derivative instead of error derivative), then the position error with step reference change would be zero, but not the velocity error (stationary error with ramp reference); this fact is stated, without proof, for brevity.

[333: dintPDplaceEN]

23:38

double-integrator PD design via pole placement

*** 

[YouTube 

This video designs a PD controller for the double-integrator plant $G(s) = 2/s^2$ to meet certain static (zero position error with reference step change, limited steady-state drift when subject to input disturbances) and dynamic (settling time less than or equal to 2 seconds, damping coefficient χ greater than or equal to 0.7) specifications.

The design is based on the final value considerations from the video [332] and on equating the characteristic equation $0 = s^2 + 2K_d s + 2K_p$ to that from a certain polynomial $0 = s^2 + 2\chi\omega_n s + \omega_n^2$ that meets the specifications.


The final part of the video simulates the calculated controller, with the correct result being verified when there is no saturation or measurement noise, which are the prior assumptions under which the theoretical calculations operated.

When saturation and measurement noise are switched on, the results are very different: derivative action noise needs to be filtered quite intensely and the transient needs to be slowed down so that the differences between 'theoretical calculations' and 'simulations closer to practice' are smaller (smaller gains both amplify noise less and saturate less, obviously). This idea is simply stated as a guideline; a redesign following it is not carried out, for brevity.


[334: dintPIDplaceEN]

21:49

double-integrator PID design via pole placement

*** 

Materiales: [[CÓD.:](#) DINTfullPIDplace.mlx] [[PDF](#)]


[YouTube 

This video designs a full-fledged PID to meet zero position error specifications both under reference changes and under input disturbances (step) of a double integrator process. This concludes the case study that began with purely intuitive tuning in the video [314], and

that the immediately preceding video [333] solved by calculating a PD because zero error was not required under input disturbances.

The final part of the video simulates the achieved performance in Matlab and incorporates saturation and measurement noise... This worsens the performance and a noise filter is tuned 'by hand', plus a 2-degree-of-freedom strategy to avoid 'derivative kick', and an antiwindup scheme is activated to avoid saturation problems in integral-action regulators. All of this is not usually the objective of a first introductory control course, but it indeed is of a 'second' course in which additional considerations are addressed so that PIDs work better 'in practice'.

Otros ejemplos Orden 2:


[335: ord2ejP]	Control de sistema de segundo orden (I): control proporcional (ejemplo resuelto Matlab)	**  19:36
Materiales: [CÓD.: EjerciciosPIDord2P.mlx] [PDF]		[YouTube ►]

Este vídeo aborda el diseño de controladores en bucle cerrado para el proceso $G(s) = 27/(s+1)/(s+9)$, centrándose únicamente en el caso del control proporcional $u = K_p \cdot e$.

En el vídeo se discute qué prestaciones de error son conseguibles con control proporcional, tanto en régimen estacionario (error de posición $e_p = 1/(1+3K_p)$ ante escalón) como en régimen transitorio (lugar de las raíces).

Se resuelve un ejemplo concreto de especificaciones de tiempo de establecimiento, error de posición y sobreoscilación que *sí* resultan factibles con control proporcional y, por contra, el vídeo termina con un conjunto de especificaciones no factibles con control proporcional, que serán obtenidas con controladores PI o PD en vídeos sucesivos.

El control PD se aborda en el vídeo [336], y el PI en el vídeo [337].


[336: ord2ejPD]	Control de sistema de segundo orden (II): control proporcional-derivado (ejemplo resuelto Matlab)	***  20:28
Materiales: [CÓD.: EjerciciosPIDord2PIandPD.mlx] [PDF]		[YouTube ►]

Este vídeo continua el caso de estudio del vídeo [335]. Al final del mismo se planteaban especificaciones no factibles con control proporcional; los tres primeros minutos del presente vídeo revisan esas ideas (comprobar no factibilidad con control P) y a continuación se pasa a resolver el problema con control proporcional+derivado, PD, de tres formas:

- Con $K_p + K_d \cdot s$, asignando dos polos reales
- Con $K_p + K_d \cdot s$, asignando dos polos complejos (oscilatorios)
- Con $K_d \cdot (s+9)$, cancelando un polo del proceso, operando con una FdT del error de primer orden.

La parte final del vídeo discute que como el derivador puro no es realizable, debe implementarse algún tipo de filtro de alta frecuencia en implementación real (no objetivo de este material).


Un control PI también puede ser una opción factible para estas especificaciones, como se discute en el vídeo [337], continuación de éste.

[337: ord2ejPI]	Control de sistema de segundo orden (III): control proporcional-integral (ejemplo resuelto Matlab)	***  12:51
Materiales: [CÓD.: EjerciciosPIDord2PIandPD.mlx] [PDF]		[YouTube ►]

Este vídeo continua el caso de estudio de los vídeos [335] y [336]. Al final de primer vídeo se planteaban especificaciones no factibles con control proporcional; el segundo vídeo discute la solución con control PD. Se recomienda la visualización previa de estos vídeos (al menos el primero).

En este vídeo se aborda el problema con control proporcional+integral, PI, utilizando forzosamente cancelación para evitar un bucle cerrado de referencia a error de orden 3.

Al final del vídeo, se compara la solución PI con las soluciones PD del otro vídeo para las mismas especificaciones.


[338: ord2ejPID]	Control de sistema de segundo orden (IV): control proporcional-integral-derivado PID (Matlab)	***  17:49
Materiales: [CÓD.: EjercicioPIDord2PIDcompleto.mlx] [PDF]		[YouTube ►]

Este vídeo continua el caso de estudio de los vídeos [335], [336] y [337]. Al final de primer vídeo se planteaban especificaciones no factibles con control proporcional; el segundo vídeo discute la solución con control PD, el tercero con control PI. Se recomienda la visualización previa de estos vídeos.

En este vídeo se aborda el problema de asignación de polos con control proporcional+integral+derivado, PID, utilizando forzosamente cancelación para evitar un bucle cerrado de referencia a error de orden 3, dado que primero se comprueba que ni P, ni PI, ni PD, pueden dar lugar a soluciones factibles para las especificaciones planteadas.

Cuestiones de realizabilidad de la derivada (necesidad de filtrado) y antiwindup integral hacen que el diseño realizado todavía no sea apto para una implementación práctica, pero la solución propuesta suele ser suficiente en una primera asignatura de introducción a los sistemas automáticos.

Caso de ejemplo ‘1er orden + integrador’, control de un eje con rozamiento


[339: pideje1]	Control de eje (1): planteamiento del problema, modelo físico y función de transferencia	**  11:52
Materiales: [CÓD.: ModelYControllerOrdIntegradorPart1.mlx] [PDF]		[YouTube ►]

Este vídeo es el primero de un caso de estudio de introducción al control mediante un modelo de un eje que gira.

En este primer vídeo se plantea el problema: las variables intervinientes (manipulada: par motor, perturbación: par resistente ejercido por una posible carga, estados: posición y velocidad, variable controlada: habrá casos de control de posición y otros de velocidad).

Luego, se explica el modelo en representación interna de orden 2 o en transformada de Laplace. Al final, es de tipo ‘primer orden’ para control de velocidad, o de ‘primer orden + integrador’ para control de posición.

La respuesta temporal en bucle abierto de dichos modelos de velocidad y posición se calcula y analiza en el vídeo [340], continuación de este caso de estudio.

[340: eje2step]	Control de eje (2): respuesta escalón velocidad y posición en bucle abierto	**  19:12
Materiales: [CÓD.: ModelYControllerOrdIntegradorPart1.mlx] [PDF]		[YouTube ►]

Este vídeo calcula la respuesta ante escalón de las salidas ‘velocidad’ y ‘posición’ de un eje sujeto a escalones de par de su entrada manipulada. El modelo tenía función de transferencia $G_{vel}(s) = 1/(Js + b)$, y $G_{pos}(s) = 1/s \cdot G_{vel}(s)$. Dicho modelado se discutió en el primer vídeo del caso de estudio, el [339].

Obviamente, las respuestas son las ‘típicas’ de primer orden (velocidad) o primer orden + integrador (posición) de modo que escribiendo $G_{vel} = K/(\tau s + 1)$ con $K = 1/b$ y $\tau = J/b$ pueden aplicarse directamente las fórmulas ‘de los libros de teoría’; de todos modos, como se dispone de Matlab, se calculan dichas respuestas mediante transformada de Laplace de modo que, de hecho, se demuestra aquí toda la teoría necesaria sin necesidad de referir a otros materiales.

[341: eje3idba] **Control de eje (3): identificación experimental escalón velocidad y posición, bucle abierto**

Materiales: [[CÓD.:](#) ModelYControl1erOrdIntegradorPart1.mlx] [[PDF](#)]

*** 15:57

[YouTube ▶]

Este vídeo discute el problema inverso al vídeo [340]: aquí obtendremos el modelo dado un supuesto experimento donde se observe la respuesta ante escalón de par.

Si la salida observada es la velocidad, ajustaremos una función de transferencia $K/(\tau s + 1)$, primer orden sin retardo.

Si la salida observada es la posición, ajustaremos una función de transferencia $K/s/(\tau s + 1)$, esto es, primer orden + integrador.

Obviamente, producirá los mismos resultados que en el vídeo de modelado, arriba referido porque se trata del mismo sistema.

[342: eje4vP] **Control de eje (4): control Proporcional de Velocidad**

Materiales: [[CÓD.:](#) ModelYControl1erOrdIntegrador.mlx] [[PDF](#)]

*** 22:56

[YouTube ▶]

Este vídeo discute el control proporcional de la velocidad de giro de un eje siendo el par la variable manipulada (ante cambios de referencia y ante perturbaciones de tipo par de carga no medible). El vídeo es parte del caso de estudio que comenzó con el vídeo [339], y modelado en [340].

Primero, se plantean las ecuaciones genéricas del diagrama de bloques, que nos valdrán tanto para control de velocidad $G_{vel}(s) = 1/(Js + b)$ como de posición $G(s) = G_{vel}(s)/s$.

Como la función de transferencia en control de velocidad es $1/(Js + b)$, primer orden, sin integrador, el control proporcional tendrá error de posición, pero cuanto más alta sea la ganancia de control, más pequeño será el error y más rápido será el transitorio.

[343: eje5vPI] **Control de eje (5): control Proporcional-Integral de Velocidad**

Materiales: [[CÓD.:](#) ModelYControl1erOrdIntegrador.mlx] [[PDF](#)]

*** 18:43

[YouTube ▶]

Este vídeo discute el control PI (proporcional-integral) de velocidad de un eje, como parte de un caso de estudio que comenzó en el vídeo [339]. El vídeo inmediatamente anterior, el [342], discutía el control proporcional. Dicho control tenía error de posición, tanto ante referencia como ante perturbaciones. Realmente se podía hacer muy pequeño con K_p grande, y también conforme crecía K_p el bucle cerrado se hacía más rápido. No obstante, si se requiere error exactamente cero o la ganancia proporcional es excesivamente alta (dando lugar a saturación o amplificación excesiva de ruido), entonces se debería recurrir a control PI, que es lo que en el presente vídeo abordamos.

Como se tiene error cero y controlabilidad total, el problema es sencillo dado que puede asignarse los polos del bucle cerrado en cualquier punto del plano complejo.

La parte final del vídeo discute la opción de ‘cancelación’: el proceso era $10/(s+1.1)$, con lo que si se utiliza un regulador PI en forma $C(s) = K_c(s+\alpha)/s$, haciendo $\alpha = 1.1$ se tiene que, ante referencia, $e(s) = \frac{1}{1+GK}r(s) = \frac{s}{s+K_c}r(s)$ la dinámica aparente es de primer orden.

*EL “truco” de la cancelación NO es válido ante perturbaciones, porque puede comprobarse que en ese caso $e(s) = \frac{G}{1+GK}d(s) = \frac{10s}{(s+1.1)(s+K_c)}d(s)$ es de segundo orden y el polo cancelado ante referencia no lo está ante perturbación.

[344: eje6pP1] Control de eje (6): control de posición, proporcional, especificaciones estáticas

Materiales: [Cód.: ModelYControl1erOrdIntegrador.mlx] [PDF]

*** 18:15

[YouTube ▶]

Este vídeo discute el control proporcional $u = Kp \cdot e$ de la posición de un eje siendo el par (motor) una variable manipulada (acción de control) y existiendo también un par de perturbación, con lo que se debe hacer frente a referencias y perturbaciones.

Este primer vídeo sobre el control proporcional discute la obtención de la función de transferencia del error en bucle cerrado y las especificaciones estáticas (error estacionario), tanto ante entradas escalón de referencia o perturbación como ante formas de onda de las mismas de tipo rampa.

La posición tiene una función de transferencia $G(s) = 1/s/(Js+b)$, cuya entrada es el par resultante (manipulado-perturbación). Se demuestra que se consigue error cero ante referencias escalón, error finito $1/Kp$ ante perturbación escalón, también error finito $1/b/Kp$ ante rampa de referencia, pero error infinito ante rampa de perturbación.

Los últimos cinco minutos del vídeo plantean que los resultados pueden extenderse a cualquier sistema ‘tipo 1’, con $G(s) = Q(s) \cdot 1/s$, esto es, con un polo en el origen. Esta es la forma en la que usualmente se explica esto en las clases de teoría de los cursos iniciales de control, aunque aquí lo hayamos motivado con el ejemplo ‘físico’ del control de posición de un eje

[345: eje7pP2] Control de eje (7): control de posición, proporcional, especificaciones dinámicas

Materiales: [Cód.: ModelYControl1erOrdIntegrador.mlx] [PDF]

*** 15:53

[YouTube ▶]

Este vídeo continua el análisis del control proporcional para un proceso de función de transferencia $G(s) = 10/s/(s+1.1)$, control de posición de un eje con par manipulado. Las especificaciones estáticas se analizaron en el vídeo [344] y aquí se analizarán las especificaciones dinámicas (oscilaciones, tiempo de establecimiento), que dependen de la posición de los polos de bucle cerrado.

En concreto, la ecuación característica era $s^2 + 1.1s + 10K_p = 0$, con lo que los polos serán $-0.55 \pm 0.5\sqrt{1.1^2 - 10K_p}$. Para K_p pequeña se obtienen dos polos reales, para K_p grande dos polos complejos (subamortiguados). La parte real de los polos no puede ser más rápida de -0.55 , con lo que existe un límite a la rapidez del tiempo de establecimiento que se puede conseguir con control P.


El vídeo detalla gráficamente el ‘lugar de las raíces’ en forma de ‘cruz’ típico de los sistemas de segundo orden sin ceros + proporcional, siendo el paso de real a imaginario en el punto medio de los dos polos del proceso.

La parte final del vídeo simula el transitorio ante referencia con distintos valores de ganancias (amortiguamiento crítico, subamortiguado y sobreamortiguado) para tener una imagen del tipo de respuesta esperable.

Un ‘problema tipo de examen’ se desarrollará en el vídeo [346].

[346: eje8pP3]

Control de eje (8): control de posición, proporcional, problema tipo concreto

***  10:48

Materiales:

[[Cód.:](#) ModelYControllerOrdIntegrador.mlx] [[PDF](#)]

[[YouTube](#) ►]

Este vídeo continúa el estudio del control proporcional en un modelo de eje con par manipulado. El detalle teórico sobre especificaciones estáticas se discutió en el vídeo [344], las especificaciones dinámicas y el lugar de las raíces en el vídeo [345]. Aquí se hace un resumen rápido y se diseña un controlador, basado en dichos vídeos, para tener:


- Error de posición estacionario igual a cero ante referencias escalón
- Error estacionario por debajo de cierta cota ante perturbaciones de par escalón
- Tiempo de establecimiento prefijado

Se obtiene una solución factible que, no obstante, es bastante oscilatoria (no se había pedido nada al respecto); si se deseara limitar las oscilaciones, entonces sería necesario usar acción derivada como se discutirá en los siguientes vídeos del caso de estudio.

Errata: las especificaciones dicen 1.5 rad/s por Nm de par de carga, pero controlando posición, las unidades correctas son 1.5 radianes por Nm.

[347: eje9pPD1]

Control PD de posición de eje (9): planteamiento general y ejemplo 1

***  10:09

Materiales:

[[Cód.:](#) ModelYControllerOrdIntegrador.mlx] [[PDF](#)]

[[YouTube](#) ►]

Este vídeo aborda el control de posición de un eje de función de transferencia $1/s/(Js+b)$ utilizando un regulador PD (proporcional-derivado). Es continuación del caso de control proporcional donde había un transitorio que no podía acelerarse más de un cierto tiempo (lugar de raíces en forma de 'cruz') y que a ganancias altas tenía oscilaciones excesivas, ver vídeo [346].


*Un control PD en la práctica necesita filtro de ruido, que no es considerado aquí para hacer más simple la metodología de diseño, ver vídeo [353].

Mediante control PD se tienen las mismas características de error estacionario que con un control proporcional, pero ahora pueden posicionarse los polos en cualquier posición arbitraria del plano complejo (controlabilidad total) porque la ecuación característica es: $1 + G \cdot (K_p + K_d s) = 0$ que, después de quitar denominadores en este ejemplo salía $s^2 + (1.1 + 10K_d)s + 10K_p = 0$ con lo que con las K_d y K_p adecuadas el denominador puede igualarse a cualquier denominador deseado.

Aquí se resuelve un primer problema, sencillo. Un segundo problema incorporando especificaciones de error estacionario ante perturbaciones se analiza en el vídeo [348], continuación de este.

[348: eje10pPD2]

Control PD de posición de eje (10): ejemplo 2

***  13:44

Materiales:

[[Cód.:](#) ModelYControllerOrdIntegrador.mlx] [[PDF](#)]

[[YouTube](#) ►]

Este vídeo aborda un segundo ejemplo de diseño de control PD para un eje cuya función de transferencia de par a posición es $10/s/(s+1.1)$. El desarrollo es similar al del vídeo [347]. Aquí se pide:

- Error de posición cero ante referencia escalón, error estacionario menor a 1.25 ante perturbación escalón unidad
- Tiempo de establecimiento menor a 2.5 segundos
- Coeficiente de amortiguamiento $\zeta > 0.75$.

Se calcula la región del plano complejo compatible con las especificaciones y a partir de ahí se calculan las constantes del regulador PD.

[349: eje11pPI1]

Control eje (11): control PI, especificaciones estáticas y dinámicas con cancelación (no funciona)

Materiales: [CÓD.: ModelYControl1erOrdIntegrador.mlx] [PDF]

18:48

[YouTube ►]

Este vídeo discute el control proporcional-integral (PI) del proceso $10/s/(s + 1.1)$ que modela la posición angular de un eje sujeto a entradas de par (manipulado o perturbación). Es un primer vídeo sobre el caso PI donde se discuten:

- especificaciones estáticas: error de posición ante referencia escalón (cero), error ante perturbación escalón (cero), error de velocidad ante referencia rampa (cero), error ante perturbación rampa (finito, constante).
- especificaciones dinámicas: ecuación característica de orden 3; con ‘cancelación’ $C(s) = K_c \cdot (s + 1.1)/s$ puede tenerse un comportamiento ante referencia de orden 2; no obstante, dicho comportamiento NO es estable para ningún valor de K_c .

El regulador PID completo con tres polos se estudiará en el siguiente vídeo [350], que abordará lugar de las raíces y en el vídeo continuación [??] que discutirá la asignación parcial de polos.

[350: eje12pPI2]

Control eje (12): control PI posición mediante lugar de las raíces, parte 1, trazado variando cero regulador

Materiales: [CÓD.: ModelYControl1erOrdIntegrador.mlx] [PDF]

21:19

[YouTube ►]

Este vídeo discute el control PI de posición de un eje argumentando las prestaciones alcanzables mediante el *lugar de las raíces*. Es continuación del vídeo [349].

En concreto se propone un control proporcional-integral en la forma $C(s) = K_c \cdot (s + \alpha)/s$, y se factoriza como $C(s) = K_c \cdot C_1(s)$ siendo $C_1(s) = (s + \alpha)/s$.

El comando de matlab rlocus(G*C_1) dibuja el lugar geométrico de las raíces de la ecuación característica $0 = 1 + K_c \cdot (G(s) * C_1(s))$ conforme varía K_c . Obviamente, lo hace ‘quitando denominadores’, esto es, dibujando las raíces de $0 = D_G(s)D_C(s) + K_c \cdot N_G(s)N_C(s)$ siendo N y D los numeradores y denominadores de proceso G y controlador C_1 indicados con el adecuado subíndice.

El vídeo dibuja el lugar de las raíces desde $\alpha = 0$ (o sea, control proporcional) hasta $\alpha = 1.1$ (cancelación del polo estable del proceso), e incluso más allá (ramas inestables). Se comprueba que el PI con $0 < \alpha < 1.1$ consigue estabilizar el bucle cerrado, pero siempre existen polos más lentos que los que podrían conseguirse con el control proporcional: para conseguir error cero ante perturbaciones escalón (necesario PI) se paga el precio de hacer la dinámica más lenta. Si esto no fuera satisfactorio, se debería recurrir al PID completo, vídeo [??].

En este vídeo no se calcula ‘hasta el final’ ningún regulador, sólo se muestra la forma del lugar de las raíces. El cálculo en sí se hará en los vídeos [??] y [??].

[351: eje13pPI3a]

Control eje (13): control PI posición asignación parcial de polos, métodos algebraicos

Materiales: [CÓD.: ModelYControl1erOrdIntegrador.mlx] [PDF]

18:58

[YouTube ►]

Este vídeo discute el diseño de un control PI (proporcional-integral) para un eje, como parte de un caso de estudio de varios vídeos. El vídeo inmediatamente anterior es el [350], que discute las especificaciones alcanzables con dicho PI (error cero ante escalón y rampa de referencia, ante escalón de perturbación) comparadas con un proporcional (el proporcional será más rápido). También presenta la forma del lugar de las raíces para diferentes posiciones del cero en $-\alpha$ del PI $(s + \alpha)/s \cdot K_c = K_p + K_i/s$. Como el PI no tiene ‘controlabilidad total’, no se puede asignar un denominador arbitrario y, por tanto, requiere de un análisis más cuidadoso el diseño de dichos controladores.

Aquí se plantea que, como el regulador tiene dos incógnitas (K_p y K_i), podría ser posible asignar dos polos, y luego ver si hay suerte o no con la posición del tercero. En efecto, ello es posible, y se hace de dos formas:

- Eligiendo un polo deseado $p_1 = -0.4 + 0.4j$ y su conjugado $p_2 = -0.4 - 0.4j$ y planteando el sistema de dos ecuaciones y dos incógnitas $D(p_1) = 0$, $D(p_2) = 0$ siendo $D(s)$ el denominador de las FdT de bucle cerrado, base de la ecuación característica, que realmente es un polinomio $D(s, k_p, k_i)$, pero una función lineal de K_p y K_i para s fijo. Esas ecuaciones se plantean en un par de líneas de la symbolic toolbox, pero el problema para hacerlo en un examen de ‘lápiz y papel’ es que los coeficientes de las ecuaciones son números complejos. No importa conceptualmente, pero el cálculo es más laborioso, por lo que se propone una forma alternativa...
- Plantear $D(s, k_p, k_i) = D_{deseado}(s)$, sustituyendo un $D_{deseado}$ arbitrario por $((s + 0.4)^2 + 0.4^2) \cdot (s + \beta)$ siendo β una tercera incógnita que nos da el tercer polo no directamente asignable. Así se puede plantear un sistema de tres ecuaciones (los coeficientes de s^2 , s y término independiente de polinomio cúbico) y tres incógnitas (K_p, K_i, β) que se resuelve fácilmente.

Existe una tercera forma de resolver el problema, que es aplicando el ‘criterio del argumento’ y ‘criterio del módulo’ de la teoría del lugar de las raíces. Como esa tercera forma es muy común en cursos introductorios de control, se le dedica un vídeo separado, ver [??].

[352: eje14pPI3b] **Control eje (14): control PI posición asignación parcial de polos, criterios módulo/argumento en lugar de las raíces**

Materiales: [[CÓD.:](#) ModelYControllerOrdIntegrador.mlx] [[PDF](#)]

*** 19:28

[YouTube ▶]

En este vídeo se hará la asignación parcial de polos en el control de un eje utilizando los criterios del ‘módulo’ y del ‘argumento’ usuales en la metodología de ‘lugar de las raíces’, para comparar la solución con la realizada de modo puramente algebraico en el vídeo [351], anterior a este en el caso de estudio.

Filtrado acción derivada:

[353: ord2ejPIDF1] **Realizabilidad del control PID: filtro de ruido en derivada, ejemplo Matlab (I)**

13:58

Materiales: [[CÓD.:](#) PIDord2ConFiltroRealista.mlx] [[PDF](#)]

[YouTube ▶]


Este vídeo usa el control PID diseñado en el vídeo [338] para discutir la invalidez en la práctica de todos los diseños de ese tipo. En efecto, comprueba que la función de transferencia entre referencia y acción de control es no realizable, porque tiene más ceros que polos (ver [267] para más detalles y ejemplos). En frecuencia, su respuesta tiende a infinito conforme crece la frecuencia.

Ninguna función de transferencia con respuesta en frecuencia infinitamente grande a alguna frecuencia (o a infinito) puede ser el resultado de un problema de control realista: será no realizable o marginalmente inestable.

Por ello, es necesario modificar el regulador PID añadiendo un filtro a la acción derivada, al menos de primer orden. En este ejemplo en particular, como los polos de bucle cerrado deseados eran -6 , se pone un filtro con polo en -60 y ganancia 1 a baja frecuencia, para limitar la amplificación del ruido de alta frecuencia en la acción de control.

Con este filtro, el bucle ya es “realizable” de modo que desde un punto de vista formal ya puede ser implementado y simulado en código o con circuitos electrónicos. No obstante, la amplificación inicial de la acción de control es 45 veces más grande que la necesaria para mantener el valor final. Eso sigue siendo “poco realista” en muchas aplicaciones prácticas. La parte final del vídeo esboza posibles modificaciones a las metodologías de diseño, y una de ellas (aumentar el filtrado teniendo en cuenta su influencia en los polos de bucle cerrado) será el objetivo del vídeo [354], continuación de éste.


*Nota: la discretización de PIDs a períodos “no demasiado pequeños” podría funcionar sin filtro de derivada, dado que el muestreo “elimina las frecuencias más allá de la frecuencia de Nyquist”. Es simplemente una puntualización, fuera de los objetivos de este vídeo, para que pienses en ello si sabes de qué estamos hablando.

[354: ord2ejPIDF2] 14:33	Control PID con filtro de derivada: diseño por asignación de polos, ejemplo Matlab (II)	*** 
Materiales: [CÓD.: PIDord2ConFiltroRealista.mlx] [PDF]		[YouTube ▶]

Este vídeo es continuación del caso de estudio de PID con filtro de ruido que se comenzó a discutir en el vídeo [353]. Los dos primeros minutos resumen las consideraciones del referido vídeo: los PIDs ideales $K_p + K_i/s + K_d s$ no son realizables, se debe filtrar al menos la acción derivada sustituyendo $K_d s$ por $K_d s/(\tau_f s + 1)$ y en el vídeo anterior se propuso un filtro 10 veces más rápido que el polo dominante deseado de bucle cerrado. El golpe de control inicial ante escalón de referencia era inaceptable.

En este vídeo, usando cancelación para mantener la dinámica observable de referencia a error de orden 2, se diseña un PID con filtro de ruido donde el propio polo del filtro es un parámetro de diseño. Con ello se consigue un regulador con mucho menos salto inicial (derivative kick) que los diseños en vídeos anteriores. A falta de un posible antiwindup integral si se considera que puede haber saturación, el regulador aquí diseñado es el más “realista” de todos (bueno, faltaría abordar, quizás, temas de discretización para su implementación por computador/PLC a período de muestreo finito, fuera de los objetivos de este vídeo):

Simulación en bucle cerrado

[355: bcode45]	Simulación de un bucle cerrado de control con ode45 (Matlab): péndulo invertido con regulador PD	***  17:45
Materiales: [CÓD.: SimulaBucleCerradoPenduloODE45.mlx] [PDF]		[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo presenta código para calcular las derivadas del estado de un bucle cerrado de control (el estado es la yuxtaposición del estado del proceso y del estado del controlador), sin necesidad de usar la interfaz gráfica de Simulink, simplemente con código Matlab ordinario.

Primero obtiene modelos en representación interna (ecuaciones de estado y de salida) del proceso controlado (péndulo en su equilibrio superior inestable), y del controlador (un

regulador PD lineal). Luego, usa esas ecuaciones en una función que calcula la derivada temporal de ambos estados.

Con `ode45` se simula el bucle resultante, y se compara con la simulación obtenida mediante el Control Systems Toolbox (comandos `feedback` y `initial`), observándose que coinciden si las condiciones iniciales están cerca del equilibrio superior (que es estable en bucle cerrado). Si las condiciones iniciales están alejadas, la simulación no lineal nos muestra que no se tiene par suficiente (por saturación) para elevar el péndulo de modo que la estabilidad es “local”; este hecho no se observa en la simulación linealizada (los modelos lineales son siempre estables “globalmente”).

► *Vídeos didácticos para Tecnología Automática:*

[Jose V. Salcedo y otros]
[Enlace]

10.7 Error en régimen estacionario en bucle cerrado

[356: errposv1]

Error de control en régimen estacionario: motivación, error de posición y velocidad, teorema valor final

***  17:24

Materiales: [[CÓD.:](#) errposveletc.mlx] [[PDF](#)]

[YouTube ►]

Este vídeo discute el concepto de error de bucle y la importancia práctica de saber su comportamiento y propiedades cuando el transitorio haya acabado (régimen estacionario o equilibrio): deberemos saber si es cero, si no lo es pero es finito (y cuánto vale), o si se hace infinito una vez las exponenciales de la respuesta se hayan extinguido.

Se presentan los conceptos de:

Error de Posición e_p : valor final del error de bucle cerrado ante referencia escalón unitario.

Error de Velocidad e_v : valor final del error de bucle cerrado ante referencia rampa de pendiente unitaria. Importante: NO tiene dimensiones de velocidad, pese a que su nombre lo sugiera; la velocidad es de la referencia.

Se utiliza el teorema del valor final para calcularlos a partir de las propiedades de la función de sensibilidad de bucle cerrado $S = 1/(1 + GK)$. Detalles sobre ese teorema valor final pueden consultarse en los vídeos [??], [133], [134] y [??].

Se demuestra que para que e_v sea finito, forzosamente S ha de tener una raíz en cero en su numerador (un *derivador* puro que convierta rampas en escalones) o, equivalentemente, $L = GK$ ha de tener al menos un polo en el origen (o sea, o lo tiene G o lo tiene K por acción integral).

La idea básica es que como $e(s) = \frac{1}{1+G(s)K(s)}r(s)$ en un bucle con proceso G y controlador K , al “quitar denominadores” para dejar la FdT de referencia a error en “bonito”, el número de polos en el origen (integradores) de $G \cdot K$ es igual al número de *ceros* en el origen (derivadores) de $\frac{1}{1+G(s)K(s)}$.

Más detalles sobre esto y generalizaciones serán objeto de futuros vídeos.

Importante: aunque aquí se desarrolla el concepto de “error ante cambios de referencia” porque es la forma usual de explicarlo en una primera asignatura de control, realmente la importancia práctica podría estar más en el error ante perturbaciones de salida, porque los errores ante referencia son conocidos, se pueden precalcular, y por tanto puede modificarse la entrada al regulador para eliminarlo; eso es lo que se denomina control con 2 grados de libertad, discutido en los vídeos [517] y [518], entre otros.

[357: errposv2]

Error bucle cerrado estacionario: error de posición y velocidad según tipo de sistema (núm. integradores)

***  14:57

Materiales: [[CÓD.:](#) errposveletc.mlx] [[PDF](#)]

[YouTube ►]

Este vídeo es continuación del vídeo [356] en el que se motivaba el problema del error estacionario en bucle cerrado, y se definían los conceptos de función de sensibilidad en bucle cerrado $S = 1/(1 + L)$, siendo $L(s) = G(s) \cdot K(s)$, de error de posición (ref- escalón, $e_p = S(0)$) y error de velocidad (ref. rampa, $e_v = \lim_{s \rightarrow 0} S(s)/s$).

Aquí se define el “tipo” de un sistema como el número de integradores (polos en el origen) de su función de transferencia, de modo que un sistema tipo m tiene una FdT que se puede escribir como $\frac{N(s)}{s^m \cdot \bar{D}(s)}$. Al quitar denominadores en la sensibilidad, el denominador de $L(s)$ pasa a ser el numerador de $S(s)$, de modo que la sensibilidad tendrá m “derivadores puros” en su numerador y, por tanto, aplicarán las ideas de valor final ante entradas escalón, rampa o polinomiales en un caso general, desarrolladas en la parte final del vídeo [135].

Aquí se insiste en los conceptos de entrada escalón o rampa en referencia (o perturbación a la salida), de modo que:

- e_p es finito para $L(s)$ tipo cero, y cero para $L(s)$ de tipo ≥ 1
- e_v es finito para $L(s)$ tipo 1, y cero para $L(s)$ de tipo ≥ 2

Las entradas polinomiales y tipos de $L(s)$ de grado arbitrario son discutidas en el vídeo [358], continuación de éste. El vídeo es interesante desde un punto de vista teórico, pero opcional si estás comenzando en estos temas, porque, desde un punto de vista práctico dado que entradas “parabólicas” o sistemas con “cinco integradores” no son usuales en las aplicaciones.

El vídeo [359] detalla algunos ejemplos de todo esto.

[358: errposv3] Error bucle cerrado estacionario ante entradas polinomiales según tipo de sistema

**** 10:55

Materiales: [CÓD.: errposvelec.mlx] [PDF]

[YouTube ▶]

Este vídeo es continuación de los vídeos [356] y [357], en el que se motivaba el problema del error estacionario, y se definían los conceptos de error de posición (ref- escalón) y error de velocidad (ref. rampa) y se calculaban en función del tipo del sistema (número de polos en el origen de $L(s) = G(s) \cdot K(s)$).

Aquí se discute el caso general, de una entrada polinomial de referencia o perturbación a la salida, y un sistema de tipo arbitrario. La idea básica es que si $L(s)$ tiene m integradores, entonces $S = 1/(1 + L)$ tiene m derivadores y aplica directamente el resultado final del vídeo [135]: el error será cero ante referencias polinomiales de grado menor a m , no cero pero finito si el grado de la referencia es m y será infinito si el grado de la entrada referencia es mayor a m .

El vídeo [359] detalla algunos ejemplos de todo esto.

[359: errposv4] Error bucle cerrado estacionario: ejemplos Matlab tipo 0,1,2 ante referencia escalón, rampa o parábola

*** 10:57

Materiales: [CÓD.: errposvelec.mlx] [PDF]

[YouTube ▶]

Este vídeo aplica la teoría de los vídeos [356], [357] y [358] a tres ejemplos desarrollados en Matlab con función de bucle abierto $L(s) = G(s)K(s)$ de distinto tipo (número de integradores): :

- Sistema sin integradores con control proporcional ($L(s)$ es tipo 0): error de posición constante, error ante rampa tiende a infinito en forma rampa.
- Sistema sin integradores con control PI ($L(s)$ es tipo 1): error de posición cero, error de velocidad finito.
- Sistema tipo 1, con regulador PID ($L(s)$ es tipo 2): error cero ante referencias escalón y rampa, error finito ante referencia parabólica (denominado en algunos textos como *error de aceleración*).

10.8 Análisis de las prestaciones de un bucle de control: respuesta en frecuencia


Nota: para un seguimiento correcto de los contenidos de las secciones 10.8.1 y 10.8.2, el lector debería tener familiaridad con los conceptos de respuesta en frecuencia de un sistema lineal invariante en el tiempo (vídeo [freqresp(10:57)]) y diagrama de Bode (vídeo [bode(09:49)]).

10.8.1 Análisis en frecuencia de bucles de control: criterio de estabilidad de Nyquist

Esta sección discute el denominado criterio de estabilidad de Nyquist, un método gráfico basado en la respuesta en frecuencia desarrollado en los años 1930, con gran importancia histórica. El criterio aplica a sistemas lineales invariantes en el tiempo en bucle cerrado; versiones más conservativas, pero que aplican a un caso general donde alguno de los elementos pueda ser no lineal, se discutirán en el vídeo [OApeqga(10:58)] (teorema de “pequeña ganancia”).

[360: nyqmo]

Criterio de estabilidad de Nyquist: Diagrama e ideas preliminares sobre estabilidad en bucle cerrado
Materiales: [NyquistMotivacion.zip]

***  07:50
[YouTube ▶]

Este vídeo motiva la idea básica del denominado criterio de Nyquist: si un bucle abierto $L(s)$ se conecta en bucle cerrado con realimentación negativa, su ecuación característica será $1 + L(s) = 0$. Por ello, los puntos con $L(s) = -1$ serán los polos del bucle cerrado.


El criterio de estabilidad de Nyquist es puramente frecuencial, intentando determinar si la imagen de la “parte derecha” (inestable) del plano complejo toca o no el punto -1 . Con un modelo, se pueden directamente calcular las raíces de $1 + L(s)$, pero se desarrollaron metodologías gráficas frecuenciales en los años 1930, que no necesitan un “modelo”, sino que se pueden aplicar a una respuesta en frecuencia experimental de un ensayo ante senoides.

El vídeo presenta un ejemplo “motivador” de lo que se busca, en Matlab. El código es correcto, pero muy poco “elegante”: realmente no se debe hacer el análisis literalmente como aparece en el vídeo, cuyo interés es sólo “motivador”. La forma correcta, con el comando `nyquist`, se aborda en los ejemplos Matlab del vídeo [362].

Nota: El lector con poco tiempo puede pasar directamente a los ejemplos de Matlab (vídeo [nyqml1(06:59)]), si no está interesado en familiarizarse con los conceptos usados en la demostración formal del criterio, esbozados muy brevemente en el siguiente vídeo [nyqte(09:52)].

[361: nyqte]

Criterio de estabilidad de Nyquist: ideas intuitivas asociadas y enunciado formal
Materiales: [Nyquist1.pdf]

****  09:52
[YouTube ▶]

La idea inicial motivada en el vídeo [360] es desarrollada y formalizada aquí.


Básicamente, el grafo de $L(j\omega)$ es una curva que delimita el borde entre las imágenes de los $L(s)$ para s estables (parte real negativa) y los $L(s)$ con $Re(s) > 0$ (inestables). Informalmente, el objetivo del criterio de Nyquist es determinar si el -1 cae “dentro” o “fuera” de esa curva, “a un lado” o “a otro lado”.

Si el proceso es realizable y estable en bucle abierto, las imágenes $L(s)$ de los puntos s del semiplano derecho deberán mapear a la zona “acotada” (no hay polos allí), “dentro” de la curva, esto es, completamente rodeada por ella. Si el proceso es inestable, deberán mapear “fuera”, en cierto sentido... formalmente, ideas de transformación conforme y principio del argumento de Cauchy servirán para, finalmente, enunciar el criterio de estabilidad completo de forma correcta: número de vueltas al -1 en sentido antihorario igual al número de polos inestables de $L(s)$ en bucle abierto. Refinamientos y discusión adicionales son abordados en el vídeo [364] tras los ejemplos Matlab.

Ejemplos Matlab básicos

[362: nyqml1]

Criterio estabilidad Nyquist: ejemplo Matlab 1 (sistemas estables en bucle abierto)

**  06:59


Materiales: [[CÓD.:](#) EjemplosNyquist1a.mlx] [[PDF](#)]

[\[YouTube ▶\]](#)

Este vídeo presenta ejemplos Matlab, usando el comando `nyquist`, de aplicación del criterio de Nyquist para determinar estabilidad en bucle cerrado de distintas configuraciones de procesos estables en bucle abierto y reguladores proporcionales o proporcional-derivativo. En el caso estable en bucle abierto, sólo debe comprobarse si el punto -1 es o no “rodeado” por la curva de Nyquist. Los resultados son idénticos a los previstos por la metodología del “lugar de las raíces”, `rlocus`, como era de esperar.

[363: nyqml2]

Criterio estabilidad Nyquist: ejemplo Matlab 2 (sistemas inestables en bucle abierto)

***  06:56

Materiales: [[CÓD.:](#) EjempNyquist1inest.mlx] [[PDF](#)]


[\[YouTube ▶\]](#)

Este vídeo aborda en caso donde se desee comprobar si determinados controladores son capaces de estabilizar en bucle cerrado a un proceso inestable en bucle abierto. En este caso, ya se debe comprobar el número de vueltas en sentido antihorario al -1 . Se confirma, también, que los resultados coinciden con los esperados al visualizar el lugar de las raíces. Se desarrollan ejemplos con sistemas con un polo y con dos polos en bucle abierto.

Refinamientos y discusión sobre el criterio de Nyquist

[364: nyqte2]

Criterio de estabilidad de Nyquist: refinamientos y discusión

****  08:44

Materiales: [Nyquist2.pdf]

[\[YouTube ▶\]](#)

Este vídeo discute las modificaciones necesarias al criterio de Nyquist para contemplar la estabilización en bucle cerrado de sistemas que, en bucle abierto, tienen polos en el eje imaginario (suele ser el caso de integradores en reguladores para corregir error de posición, o integradores/osciladores puros en modelos mecánicos sin rozamiento). También se discute el caso de realimentación positiva donde, como era de esperar, se debe utilizar `nyquist(-L)` para evaluarlo.

La segunda mitad del vídeo discute las principales ideas que hacen importante históricamente en la teoría de control el criterio:

- $L(j\omega)$ puede ser obtenida experimentalmente, sin un modelo $L(s)$ de polos y ceros.
- La estabilidad en bucle cerrado de sistemas con retardo o resultantes de ecuaciones en derivadas parciales (de orden teóricamente infinito, muchas veces aproximado por un orden muy alto; ver ejemplo en intercambiador en vídeo [184]) puede ser determinada de forma exacta mediante el criterio de Nyquist, mientras que la evaluación de ceros de $1 + L(s)$ requiere aproximaciones polinomiales de $L(s)$ con el comando `pade` en el caso del retardo, o con elementos finitos en el caso de derivadas parciales (vídeos [78], [278]).
- La distancia del grafo de $L(j\omega)$ al -1 determina lo que se denominan “márgenes de robustez” (evaluar el “riesgo” de que el bucle cerrado sea inestable ante los siempre presentes errores de modelado). Ello dio lugar a técnicas de diseño en frecuencia de PID (o sus primos hermanos que denominaban redes de adelanto/atraso) en los años 1940 considerando no sólo prestaciones sino también tolerancia a error de modelado (robustez), luego al “quantitative feedback theory” en los años 1960, e inspiraron el enfoque frecuencial del control robusto de los años 1980, que se discute en detalle en la Parte sobre control robusto de estos materiales.

[365: nyqml3] Criterio de estabilidad de Nyquist con polos parte real cero y retardo: ejemplo Matlab *** 08:53
Materiales: [[CÓD.:](#) EjemploNyquist2.mlx] [[PDF](#)] [YouTube ►]

Este vídeo presenta ejemplos Matlab de sistemas con polos de parte real cero en bucle abierto: un regulador PI y un doble integrador (masa sin rozamiento). La idea se basa en “perturbar” un poquito los polos para moverlos a parte real diferente de cero... se cambia $1/s$ por $1/(s + 0.001)$. El vídeo [366] ilustra que si se hubiese cambiado a $1/(s - 0.001)$ las conclusiones de estabilidad hubiesen sido las mismas.

El vídeo también discute un ejemplo de aplicación del criterio de estabilidad de Nyquist a un sistema con retardo. El bucle cerrado tiene infinitos polos, y no puede ser representado como una función de transferencia polinomial-racional usual, pero eso no importa: el criterio de Nyquist no tiene problema para determinar su estabilidad, mientras que un cálculo numérico de polos tendría que usar [pade](#) y el resultado sería únicamente aproximado.

[366: nyqml4] Criterio de estabilidad de Nyquist con polos parte real cero: ejemplo Matlab (2) *** 05:58
Materiales: [[CÓD.:](#) EjemploNyquist2b.mlx] [[PDF](#)] [YouTube ►]

Este vídeo presenta ejemplos Matlab de sistemas con polos de parte real cero en bucle abierto: un regulador PI y un doble integrador (masa sin rozamiento). En particular, discute un poco más la opción de perturbar ligeramente esos polos que ya se comentó en el vídeo [365]. Aquí se comprueba que si se hubiesen perturbado “hacia el lado inestable” las conclusiones hubieran sido las mismas.

Nota: Dado que el criterio de Nyquist determina la estabilidad de un bucle cerrado en función de las veces que rodea el diagrama de bucle abierto al -1 , la distancia al punto -1 en el plano complejo está relacionada con la tolerancia a errores de modelado (robustez) del diseño. Esta idea dió lugar, históricamente, a lo que se conocen como márgenes de estabilidad, que se discuten en este documento en la Sección 26.1.


10.8.2 Análisis en frecuencia de bucles de control: respuesta en frecuencia en bucle cerrado

[367: rfbc] Análisis de respuesta en frecuencia de bucles cerrados de control (monovariable) *** 16:46
Materiales: [rfbc.pdf] [YouTube ►]

En este vídeo se discuten las características principales de la respuesta en frecuencia de las funciones de transferencia entre referencia (entrada en bucle cerrado) y las salidas: error, salida medida y acción de control. Obviamente, se presupone la estabilidad de dicho bucle, bien por que se evalúan los polos, bien por el criterio de Nyquist puramente frecuencial discutido en los vídeos [360] y [364], entre otros.

El bucle cerrado analizado es el bucle convencional de un grado de libertad $y = Gu$, $u = Ke$, $e = (r - y)$, con lo que $e = Sr = \frac{1}{1+GK}r$, $u = KSr = \frac{K}{1+GK}r$, $y = Tr = \frac{GK}{1+GK}r$. Se discute la (poco clara) relación de los parámetros de la respuesta en frecuencia con la rapidez (tiempo de subida) y la sobreoscilación de la respuesta temporal.

También se discute, por último, la interpretación de dichas funciones de transferencia en problemas de rechazo de perturbaciones, para los que el enfoque frecuencial es, en caso de ruidos coloreados, más informativo que las características de una respuesta ante escalón que la perturbación, en principio, no presenta.

[368: rfbcm] **Análisis de respuesta en frecuencia de bucles de control:** ***  14:22

Ejemplo MATLAB

Materiales: [**Cód.:** anfrbc.mlx] [**PDF**]

[YouTube ▶]

En este vídeo se comparan las respuestas temporales (step) y en frecuencia de bucle cerrado de dos reguladores (PID y PD, diseñados por prueba y error) analizando los elementos de S , T y KS descritos en el vídeo [367]: valores a bajas y altas frecuencias, anchos de banda, picos de resonancia.

Otro ejemplo de análisis frecuencial en bucle cerrado en un contexto de control por modelo interno (IMC) aparece en el vídeo [396], para los lectores familiares con dicha metodología; la respuesta en frecuencia de bucles de control es la idea básica fundamental en lo que se denomina \mathcal{H}_∞ loop shaping (sensibilidad mixta, etc.), por lo que las ideas de este vídeo se usarán con profusión en dicha técnica, aunque no se pone por el momento enlaces a los mismos dado que requieren una adecuada introducción.

Un análisis de la respuesta en bucle cerrado de un proceso “incierto” (usando la incertidumbre **ultidyn** de la robust control toolbox) se aborda en el vídeo [854].

Capítulo 11

Control por computador: selección de período de muestro y discretización

Este capítulo aborda cómo utilizar técnicas discretas para controlar procesos físicos que, obviamente, son de tiempo continuo: se discute el proceso de muestreo y reconstrucción de señales, la discretización de procesos lineales y la selección de un período de muestreo razonable para una aplicación de control.

11.1 Muestreo y reconstrucción de señales analógicas

[369: myzoh] Muestreo y Reconstrucción de Señales, comentarios sobre aplicación a control

Materiales: [MuestreoYReconstr.pdf]

**  25:36

[YouTube ▶]

En este vídeo se definen los conceptos de muestreo, período y frecuencia de muestreo. Se enuncia el teorema de Shannon (señales de ancho de banda finito) y la fórmula de reconstrucción ideal.

Se discute el concepto de *aliasing* (traslado de frecuencia) y se ponen ejemplos.

Se plantea el concepto de “retenedor de orden cero” y se esboza que hay retenedores de orden superior en la literatura, pero de poco interés en la práctica.

Se discuten las consecuencias de la no idealidad del muestreo y reconstrucción: la reconstrucción de Shannon no es causal y necesita memoria infinita, las señales físicas tienen ruido de medida de alta frecuencia (se necesita filtro *anti-alias*, y el retenedor de orden cero (ZOH) introduce un retraso de medio período o de un período si se filtra (*filtro de reconstrucción*) para eliminar los armónicos de alta frecuencia de la señal de salida (producidos por los cambios “instantaneos” en escalón del ZOH).

Se pone un ejemplo de filtros Butterworth analógicos de primer y segundo orden para antialias y reconstrucción en una aplicación de registro y reproducción de audio digital a 48 KHz.


[18:20] A partir de este instante, en el vídeo se discute la aplicabilidad de las nociones anteriores a problemas de control. Se argumenta que es sólo “orientativa” dado que los filtros introducen dinámica adicional, que los modelos ayudan a predecir componentes de la respuesta del sistema sin necesidad de medirlo y que cerrar un bucle de control cambia el espectro en frecuencia de las señales en bucle cerrado (desplazándolo hacia la alta frecuencia, efecto “*cama de agua*”). Por ello, la selección del período de muestreo más adecuado para un problema de control requiere consideraciones adicionales a las ideas Nyquist-Shannon-alias que son básicamente, las únicas que se consideran en otras aplicaciones de procesamiento digital de señal.

Bajo determinadas suposiciones estadísticas, la selección de período de muestreo podría basarse en un criterio de mínima varianza del error entre la señal y su estimación. Si eres estudiante avanzado o profesor en la materia, podría interesarte el caso de estudio de los vídeos [253], [254] y [255].

11.2 Control por computador: problema a resolver

[370: cpcp] Control por computador: planteamiento del problema y opciones de discretización

Materiales: [PlantCPC.pdf]

***  13:16

[YouTube ▶]

En este vídeo se plantea qué es un bucle de control por computador: señales intervinientes continuas (perturbaciones, objetivos de control) y discretas (secuencia de medidas y acciones de control).

Se describen las posibles estrategias para abordarlo:

1. Control muestreado (sistemas híbridos continuos/discretos), formalmente la mejor opción pero también la más compleja.
2. Diseño en tiempo continuo + Discretización posterior de reguladores continuos (aproximada)

3. Discretización de procesos (exacta en caso lineal con retenedor) + diseño posterior de reguladores en tiempo discreto.

La primera y la última de las opciones garantizarían estabilidad sea cual sea el período de muestreo. La discretización de reguladores continuos sólo garantiza estabilidad cuando el período es muy pequeño (tendiendo a cero).

Se discuten determinadas recomendaciones en algunos casos, aunque si el período es pequeño todo resulta parecido y la elección de unas u otras opciones es cuestión de preferencia del diseñador; no así con períodos grandes, en general.

11.3 Discretización de procesos lineales (para diseño posterior de reguladores discretos)

11.3.1 Discretización exacta de efecto acción de control (zoh)

[371: disc1]

Discretización de sistemas dinámicos lineales en representación interna (teoría: exponencial de matriz)

Materiales: [discretizaV.pdf]

***  20:08

[YouTube ▶]

En este video se aborda la discretización de sistemas lineales en representación interna. La idea básica es la que se plantea en el vídeo [??], donde se usa para calcular respuesta escalón con exponencial de matriz... y un computador simplemente genera un tren de escalones, manteniendo la salida de su tarjeta de conversión digital→analógico mientras se calcula la siguiente. Se recomienda visualizar el vídeo referenciado previamente a éste.

- Se presenta el bucle de control por computador, donde el proceso es representado con: →[Retenedor]→[Proceso]→[Muestreador].
- Se presentan las representaciones normalizadas en variables de estado discretas/continuas/muestreadas.
- Se describe la solución de $\dot{x} = Ax$ como $e^{At}x(0)$, introduciendo la exponencial de una matriz (para más detalle, visionar los vídeos de teoría [162] y [163], o el ejemplo de Matlab [164]).
- Se modela al retenedor de orden cero (ZOH) con un estado aumentado $\dot{u} = 0$. Es un caso particular del régimen estacionario (constante) discutido en el vídeo [165].
- Se presenta la fórmula exponencial de la discretización ZOH.
- Se incide en que es una fórmula *exacta* para este caso de discretización de procesos lineales, aunque en discretización de reguladores (vídeo [374]) o procesos no lineales no puede aplicarse, y se necesita integración numérica (aproximada).

[372: disc2]

Discretización de sistemas dinámicos lineales (ejemplo matlab)

Materiales: [Cód.: ejemplodiscretiz.m] [PDF]

*** / ****  19:21

[YouTube ▶]

En este vídeo, se presenta código de Matlab discretizando un proceso de segundo orden (masa-muelle-amortiguador), con la fórmula exponencial discutida en el video [371]. De hecho, el comando `c2d` de Matlab hace los mismos pasos internamente si se le da la opción `'zoh'`, ver documentación del mismo en la ayuda de Matlab para más detalles. **

Se superponen la simulación discretizada (sólo en instantes de muestreo) con la simulación continua (bueno, de hecho, aproximada por una simulación discreta con paso de integración/período de muestreo menor). **** ☒

Se presenta el concepto de oscilaciones ocultas si el período de muestreo no está correctamente elegido. ****

Nota: El comando `sdlsim` es el que Matlab tiene incorporado para realizar simulaciones de controladores discretos con procesos continuos y retenedor zoh; en este vídeo no se utiliza, por cuestiones didácticas para presentar cómo podría hacerse sin él (esto es, con simulación discreta a período más pequeño). En los vídeos [389] y [390] se detalla un caso de estudio sobre el uso de dicho comando; en el [382] también se utiliza.

11.3.2 Discretización por interpolación de perturbaciones con retenedor orden 1

[373: dfbh] Discretización de sistemas lineales por interpolación entre muestras (foh)

Materiales: [DiscrFOH.pdf]

**** 13:32

[YouTube ▶]

Este vídeo plantea el problema de discretizar la respuesta ante perturbaciones. Las perturbaciones no son constantes entre muestras como las acciones de control (variables manipuladas).

El vídeo aborda la discretización de sistemas continuos en representación interna.

Si se asume que la interpolación lineal entre muestras es una buena aproximación de las mismas, lo cual es cierto si el período de muestreo es suficientemente pequeño, entonces puede discretizarse usando una variación de la fórmula exponencial de matriz vista en el vídeo [371].

Esta fórmula, así como las manipulaciones para obtener una representación interna normalizada del resultado son la parte principal del vídeo. Al final, se presenta un ejemplo $G(s) = 1/s$ observándose que coincide con la discretización de Tustin (eso no ocurre con otras $G(s)$, es “casualidad” en el sentido de que sólo coinciden “Tustin” y “foh” con el integrador puro).

El comando Matlab es `c2d(G,T,'foh')`, donde ‘foh’ indica first-order-hold, retenedor de orden 1 *no causal* (para saber el valor entre muestras se debe saber el valor de la entrada en el siguiente período por lo que en tiempo continuo es “no causal”; en discreto, una vez se ha tomado la muestra de la entrada, ya puede reconstruirse el pasado entre esa muestra recién tomada y la anterior para simular la ecuación diferencial, por lo que la discretización **sí** es “causal” y, por tanto, realizable).

A períodos muy pequeños, realmente tampoco hay tanta diferencia con la suposición ‘zoh’, pero formalmente es interesante cuando el período no puede reducirse todo lo que sería deseable. De todos modos, el uso de ‘foh’ es poco frecuente.

11.4 Discretización aproximada de filtros y reguladores continuos

[374: dre1] Discretización de reguladores continuos en Función de Transferencia, conceptos básicos

Materiales: [DiscReg1.pdf]

** 10:31

[YouTube ▶]

En este vídeo se plantea el problema de la discretización aproximada de reguladores y se presentan las fórmulas de cambio de variable para discretizar reguladores continuos prediseñados en representación de función de transferencia.

Se presentan las aproximaciones de:

- Euler hacia adelante: $s = (z - 1)/T$,
- Euler hacia atrás: $s = (1 - z^{-1})/T$,

- Tustin (bilineal): $s = 2/T * (z - 1)/(z + 1)$.

Otras aproximaciones, así como el análisis de estabilidad y en frecuencia de los reguladores $K(z)$ resultantes se dejan para otros materiales.

[375: dre2e]

Discretización de reguladores, método de Euler: no preservación de estabilidad y respuesta en frecuencia

11:01

Materiales:

[DisRegStab2Euler.pdf]

[YouTube ▶]

En este vídeo se analiza la preservación de la estabilidad y la respuesta en frecuencia de los reguladores continuos discretizados mediante el método de Euler. La conclusión es negativa: el semiplano estable en el dominio de Laplace *no* mapea al círculo de radio 1 en el plano Z con lo que reguladores estables pueden discretizar como inestables (con Euler hacia adelante), o reguladores inestables pueden discretizar como estables (con Euler hacia atrás). También existe una importante distorsión de la respuesta en frecuencia, dado que cuando la variable $z = e^{j\omega T}$ recorre la circunferencia unidad, la variable s no recorre el eje imaginario, sino un círculo de diámetro $2/T$ (revisar los vídeos [192] y [200] para recordar conceptos de respuesta en frecuencia continua y discreta, respectivamente).

[376: dre2t]

Discretización por Tustin: preservación de estabilidad y respuesta en frecuencia (distorsionada)

22:50

Materiales:

[DisRegStab2Tustin.pdf]

[YouTube ▶]

En este vídeo se analiza la preservación de estabilidad y respuesta en frecuencia de reguladores continuos $K_c(s)$ discretizados mediante la aproximación bilineal (de Tustin). Los resultados parecen ser más prometedores que en el caso Euler discutido en el vídeo [375]:

- La discretización $K_d(z) = K_c(\frac{2(z-1)}{T(z+1)})$ es estable en tiempo discreto sí y sólo si $K_c(s)$ es estable (en tiempo continuo).
- El eje $\pm j\omega$ mapea a la circunferencia unidad: todo lo que se “ve” en la respuesta en frecuencia continua $K_c(j\omega)$ se “ve” en la respuesta en frecuencia discreta $K_d(e^{j\omega_d T})$.

Pese a ser bastante más exacta que las aproximaciones de Euler, no obstante, el uso de la discretización de Tustin plantea aún dos inconvenientes:

- Existe distorsión en la respuesta en frecuencia $\omega_d \neq \omega$. En concreto $\omega_d = \frac{2}{T} \operatorname{atan}\left(\frac{\omega T}{2}\right)$. La distorsión sólo está por debajo del 3% a frecuencias ω por debajo del 20% de la frecuencia de Nyquist $\omega \leq 0.2\omega_{Nyquist} = 0.2\pi/T$.
- La supuesta “exactitud” no lo es tanto, porque no tiene en cuenta la presencia de la planta controlada: la preservación de estabilidad en *bucle abierto* de $K_d(z)$ **no garantiza la estabilidad en bucle cerrado** de $K_d(z)$ junto a $GD = c2d(G, T, 'zoh')$... hay plantas en las que un regulador discretizado por Euler al mismo período funciona mejor que la discretización de Tustin, pese a que intuitivamente pudiera creerse que eso no podría suceder. Obviamente, en problemas de “filtrado” donde no hay realimentación (bucle abierto), la estabilidad del filtro resultante sí está garantizada, y a esto es lo que se refiere mucha literatura de procesamiento de señal; en control dicha garantía no existe (opciones con garantía de estabilidad en bucle cerrado se discuten en el vídeo [380]).

Nota: La selección de período de muestreo para discretización de reguladores continuos se aborda en la Sección 11.6.1, cuyo visionado podría ser aconsejable en este momento.

[377: dretg]

Discretización de reguladores: modificaciones para período de muestreo grande

10:45

Materiales:

[DiscRegContTgrande4.pdf]

[YouTube ▶]

Cuanto menor es el período de muestreo para la discretización de reguladores, menor error de discretización se comete en la aproximación de un regulador continuo. En este vídeo se discuten las acciones a tomar si el período de muestreo no es todo lo pequeño que se desearía para que dicho error de discretización sea despreciable.

Básicamente, se argumenta que a período alto la discretización de un regulador $K_c(s)$ diseñado para $G(s)$ no funciona bien, y que diseñándolo para $\hat{G}(s) := \text{d2c}(\text{c2d}(G, T, 'zoh'), 'tustin')$ (esto es, para la *continuización* por Tustin de la *discretización* exacta ZOH de la planta), el resultado garantiza al menos estabilidad en bucle cerrado discreto. Esta idea, de hecho, puede ser en sí misma considerada como una metodología de diseño de reguladores digitales y, dada su importancia, es ampliada en los vídeos [380] y [381], cuyos materiales se aconseja visionar en este momento.

Por último, se argumenta que la multiplicación por un cero de fase no mínima $\hat{G}(s) = G(s) \cdot (1 - \frac{T}{2}s)$ produce un resultado parecido al de $\text{d2c}(\text{c2d}(\dots))$ anterior y resulta mucho más rápido/cómodo de calcular o comprender. Se pone un ejemplo numérico comparando las dos opciones para \hat{G} .

Nota: Hay una errata en la transparencia 6 en el vídeo, donde dice $(1-s)$ debe ser $(1-0.05s)$, ya corregida en el PDF de materiales.

11.4.1 Discretización (aproximada) de reguladores en representación interna

Los reguladores complejos/multivariables es preferible manejarlos mediante representación interna que mediante matriz de transferencia para su implementación (realización). Por ello esta sección discute la discretización de dichos reguladores continuos cuando lo que se dispone es una representación interna, sin necesidad de pasarlos a función (o matriz) de transferencia.

[378: drsse] Discretización de reguladores continuos en representación interna: métodos de Euler
**** 11:00

Materiales: [DisRegssEuler.pdf]

[YouTube ►]

En este vídeo se discute la discretización de un regulador $ss(A_c, B_c, C_c, D_c)$ para obtener las matrices que lo describen en discreto $ss(A_d, B_d, C_d, D_d, T)$ a partir de las continuas utilizando el método de Euler hacia adelante y hacia atrás. En el segundo caso, es necesario un cambio de variables $\xi(k) = x(k-1)$ para obtener una representación interna normalizada. El significado físico de las variables de estado se conservaría (con matices) si existiera, aunque en muchos reguladores los estados no tienen un significado físico claro, con excepción de los reguladores basados en observador.

[379: drsst] Discretización de reguladores continuos en representación interna: método de Tustin
**** 14:39

Materiales: [DiscRegssTustin.pdf]

[YouTube ►]

Este vídeo discute el mismo problema que el vídeo [378] pero aplicando la discretización bilineal (Tustin). El cambio de variable necesario para transformar a representación interna ya no depende sólo del estado, sino que es $\xi(k) = (1 - A_c T/2) \cdot x(k) - B_c T/2 \cdot e(k)$, dependiente de la entrada al regulador.

En las conclusiones del vídeo se discute la estrecha relación de la discretización de Tustin y la discretización trapezoidal (foh causal) discutida en el vídeo [373]. Coinciden para integrador puro, y 'Tustin' es la aproximación de Padé de primer orden de 'foh'. Por ello, con polos lentos, Tustin y foh producen prácticamente el mismo resultado (no así con polos rápidos). Sigue siendo importante recalcar que *no se garantiza la estabilidad en bucle cerrado* al ser todo aproximaciones.

11.5 Diseño de reguladores discretos por “continuización” con garantía de estabilidad

[380: d2c]

Diseño de reguladores discretos por continuización (con garantía de estabilidad)

*** 11:00

Materiales:

[Continuizacion.pdf]

[YouTube ▶]

En este vídeo se presenta cómo utilizar técnicas de tiempo continuo para diseñar reguladores discretos con garantía de estabilidad en bucle cerrado.

La idea básica es que, dado $G_d(z)$, puedo obtener un proceso “continuizado” con la transformación bilineal (Tustin), $\tilde{G}_c(s) := G_d(\frac{1+Ts/2}{1-Ts/2})$, siendo T el período de muestreo al que opera $G_d(z)$. Si diseñamos un regulador $K_c(s)$ que estabilice $\tilde{G}_c(s)$, entonces cualquier función de transferencia de bucle cerrado (por ejemplo, de referencia a error $S = 1/(1 + K_c(s)\tilde{G}_c(s))$) será estable y, si transformamos \tilde{G}_c a discreto por Tustin recuperamos $G_d(z)$, y si transformamos $K_c(s)$ a discreto por Tustin entonces obtendremos un regulador $K_d(z)$ que garantizará que $S = 1/(1 + K_d(z)G_d(z))$ es estable (y cualquier otra función de transferencia de bucle cerrado). La demostración de esto (breve) se encuentra desarrollada en el vídeo [381], mientras que en este se discute su utilización y se razona de forma “intuitiva” sobre el tema.

Esto permite usar cualquier técnica continua y a cualquier período de muestreo sin riesgo de que la implementación discreta sea inestable, lo cual no ocurriría con la discretización de reguladores “aproximada” discutida en otros materiales. La deformación en frecuencia, no obstante, hace que sea necesario comprobar otras prestaciones (tiempo de establecimiento, sobreoscilación, etc.) porque sólo la estabilidad está garantizada.

Un ejemplo Matlab de aplicación de estas ideas se detalla en el vídeo [382].

[381: d2cp]

Diseño de reguladores discretos por continuización, con garantía de estabilidad: demostración

*** 06:38

Materiales:

[ContinuizProof.pdf]

[YouTube ▶]

El vídeo anterior [380] presentaba la motivación y una discusión “intuitiva” sobre cómo utilizar la transformación bilineal para el diseño de reguladores discretos mediante “continuización”. Este vídeo presenta la demostración detallada del resultado fundamental: si se dispone de un modelo discreto $G_d(z)$, y se diseña $K_c(s)$ de modo que estabiliza al proceso continuo $\tilde{G}_c(s) := G_d(\frac{1+Ts/2}{1-Ts/2})$ en bucle cerrado, entonces el regulador discreto $K_d(z) := K_c(\frac{2}{T} \frac{z-1}{z+1})$ estabiliza a $G_d(z)$.

[382: d2cml]

Diseño discreto de reguladores por continuización (transf.bilineal): ejemplo MATLAB

*** 12:35

Materiales:

[Cód.: continuizml.mlx] [PDF]

[YouTube ▶]

En este vídeo se presenta un ejemplo Matlab donde se aplican las ideas del vídeo anterior [380], cuya visualización previa se aconseja, usando los comandos c2d , d2c para ello. El objetivo es controlar una planta continua, que se ha discretizado de forma exacta (zoh) con un controlador digital (aunque el procedimiento es válido para cualquier modelo $G_d(z)$ de la planta a controlar, por ejemplo, obtenido por identificación).

Para diseñar el controlador discreto, se continua la planta (obteniendo un segundo modelo continuo puramente “instrumental”, diferente al original –esto es intencionado, la diferencia es consecuencia de que la continuización por Tustin no “invierte” la discretización zoh, obviamente–), y se diseña un regulador para él, que luego se discretiza. Todo funciona

bien cuando el período es pequeño. A períodos de muestreo grande, aunque la preservación de estabilidad está garantizada, existe una “distorsión” de la respuesta temporal significativa que, además, presenta importantes oscilaciones ocultas en la respuesta intermuestreo (que se ha evaluado utilizando el comando `sdl sim`; este comando se discute más detalle en el vídeo [390]).

Nota: realmente, el regulador que se discute en este vídeo es un PID sintonizado “manualmente” por prueba y error. Esta sintonía se hace en el dominio continuo, para demostrar la preservación de estabilidad al pasar a discreto... ¡dado que es por “prueba y error” podría haberse, directamente, sintonizado manualmente examinando la respuesta en el tiempo discreto, sin necesidad de “continuizar” el proceso! De hecho, esa sería la recomendación, sobre todo a períodos grandes, debido a la distorsión que induce la transformación.

Nota: Esta técnica preserva el “pico” de la respuesta en frecuencia (norma \mathcal{H}_∞) de un sistema lineal, aunque en la continuización y la discretización sea a frecuencias diferentes. Por ello, una de las metodologías a las que se adapta mejor es al control óptimo \mathcal{H}_∞ de procesos discretos que se discute en el Capítulo 19.

11.6 Selección de período de muestreo en aplicaciones de control

[383: selpm] Selección de período de muestreo: consideraciones para control

Materiales: [SelMuestreoControl1.pdf]

***  16:03
[YouTube ▶]


En este vídeo se revisan rápidamente las ideas básicas del vídeo [369], pero se argumenta que el objetivo de control no es “reconstruir” señales analógicas con “alta fidelidad” sino “reducir los componentes de baja frecuencia del error”.

Se discuten reglas genéricas, heurísticas, de selección de período de muestreo basado tanto en polos de bucle abierto poco amortiguados como en los polos de bucle cerrado y su respuesta asociada (tiempo de establecimiento, frecuencia propia, frecuencia de resonancia, tiempo de subida) y se dan fórmulas que podrían suponer una primera elección bastante razonable, a expensas de posibles modificaciones debidas a tiempo de cómputo o carga de red, ruido de medida, etc.

En este vídeo, basándose en ideas Nyquist/Shannon, se dice “a ojo” que se debería muestrear como poco a dos o cuatro veces la frecuencia de resonancia; en el vídeo [386] esta idea se refina, proponiendo una frecuencia de muestreo bastante mayor (período menor) si se desea conservar las prestaciones de un regulador continuo prediseñado que no se desea cambiar.

[384: Tpeq] Control digital: problemas por período de muestro demasiado pequeño

Materiales: [ProblemaTpeq.pdf]

***  11:00
[YouTube ▶]

En este vídeo se presentan los inconvenientes asociados a un período de muestreo demasiado pequeño en una aplicación de control:

- Las mejoras se hacen inapreciables a partir de un cierto valor de T suficientemente pequeño en la mayoría de aplicaciones,
- Aumenta la carga computacional y la saturación de red,
- La limitación de ancho de banda de sensores y actuadores y la cuantización hace que en la práctica no se obtenga más información “útil” por muestrear muy rápido,
- Hay problemas en la identificación (muchas muestras consecutivas prácticamente iguales: $y_{k+1} = y_k$),

- Se necesitan cálculos con aritmética de más bits de precisión en las funciones de transferencia (o implemmentaciones numéricamente adecuadas como "Second Order Sections", SOS).

[385: selT2]
Selección de período de muestreo en control: escenarios especiales

CE
14:15

Materiales: [SelMuestreo2.pdf]
[YouTube ▶]

En este vídeo se discuten brevemente escenarios "especiales" donde las reglas de "andar por casa" consideradas en el vídeo [383] podrían necesitar modificaciones; también la selección de período de muestreo para utilizar reguladores continuos discretizados (aproximadamente) podría requerir consideraciones especiales que se discutirán de modo específico en otros materiales, ver vídeos [386], [387] y [388].

En concreto, se discuten escenarios con ruido de medida elevado, proponiéndose la posibilidad de un sobremuestreo y filtrado digital a frecuencia mayor antes de devolver resultados a una frecuencia menor para, por ejemplo, un PID de control. Esto es similar a lo que realiza el comando **decimate** de la Signal Processing toolbox de Matlab.

En caso de actuadores "frágiles" podría también estar indicado un muestreo de control a mayor frecuencia y, a lo mejor, algún tipo de filtrado o suavizado en la propia electrónica del actuador.

Por último, se considera que en una estructura multivariable podría haber bucles con distintos requerimientos de control y distintos períodos de muestreo asociados como, por ejemplo, con control en cascada.

En general, todas esas situaciones anteriores podrían dar lugar a *escenarios multifrecuencia* aunque, si hay separación de escalas de tiempo o todos los períodos de muestreo son muy pequeños, son en la práctica transparentes al usuario/diseñador de reguladores que cree que trabaja con PID's continuos o monofrecuencia.

Sólo en algunas situaciones, relativamente infrecuentes en aplicaciones industriales, podría ser necesaria la teoría de control multifrecuencia específica, que trata formalmente estos escenarios, discutida en otros materiales.

Bajo determinadas suposiciones estadísticas, la selección de período de muestreo podría basarse en un criterio de mínima varianza del error entre la señal y su estimación. Si eres estudiante avanzado o profesor en la materia, podría interesarte el caso de estudio de los vídeos [253], [254], [255] y [256].

11.6.1 Selección de período de muestreo para discretización de reguladores continuos

[386: dreslt]
Selección de período de muestreo para discretización de reguladores continuos

🔔
11:00

Materiales: [DiscRegContElegirT3.pdf]
[YouTube ▶]

En este vídeo se discuten criterios específicos para la selección del período de muestreo en reguladores digitales obtenidos a partir de discretización de reguladores continuos preexistentes.

Primero, se revisan brevemente las fórmulas de discretización (vídeos [374],[375]), preservación de estabilidad (no en Euler, sí en bilineal-Tustin), y distorsión de respuesta en frecuencia (vídeo [376]).

Luego, se discute que si $G(j\omega)K(j\omega) \approx -1$ el sistema está cerca de que la función de sensibilidad (FdT de referencia a error de bucle) $S = 1/(1 + GK)$ está cerca de tener

polos que cruzan el eje imaginario (límite de estabilidad, $1 + G(j\omega)K(j\omega) = 0$). Para más detalles sobre el significado de la respuesta en frecuencia de las funciones de bucle cerrado, se aconseja visionar el vídeo [367].

Como por debajo de $0.2\omega_{Nyquist}$ la distorsión es poca, se propone, orientativamente, que la frecuencia ω_{PICO} donde S tiene un pico de resonancia (mínimo módulo de $1 + G(j\omega)K(j\omega)$) sea, como mucho la quinta parte de la frecuencia de Nyquist, esto es, $T \leq \pi/(5\omega_{PICO})$. Esta cota de T es algo menor, pero parecida, a la cota $T \leq \pi/(5\omega_{0dB})$, siendo ω_{0dB} la frecuencia más alta donde $|G(j\omega)K(j\omega)| = 1$ (frecuencia de cruce, crossover frequency) que otros autores proponen.

Esta cota refina la idea, “a ojo”, intuitiva, que se lanzó de forma preliminar en el vídeo [383] donde se argumentaba que, por cuestiones relacionadas con Nyquist/Shannon, debería muestrearse al doble o cuádruple de las frecuencias de resonancia de bucle cerrado... la frecuencia de muestreo propuesta en el presente vídeo [386] para preservar las prestaciones continuas sin modificar el diseño de K (simplemente discretizarlo) es de 10 veces la frecuencia de resonancia de la sensibilidad, motivado no sólo por cuestiones de ancho de banda de Nyquist, sino también por calidad de la aproximación.

[387: drem] Discretización de reguladores continuos: ejemplo Matlab Euler, Tustin, FOH

Materiales: [[CÓD.:](#) testdiscretiz.mlx] [[PDF](#)]

*** 17:02

[YouTube ▶]

En este vídeo se presenta el diseño de un PID para la planta $G(s) = 3/(s+1)/(s+2)$, en concreto $K(s) = 1.8 + 1.2/s + 0.25s/(0.05s+1)$. La respuesta se considera satisfactoria, y se comparan las prestaciones en bucle cerrado de diferentes discretizaciones (Euler hacia adelante, hacia atrás, Tustin y FOH) y a diferentes períodos de muestreo.

También se comparan las simulaciones con la respuesta obtenida cerrando el bucle $G(s)(1 + \frac{T}{2})K(s)$ que, como se discutió en el vídeo [377], aproxima de modo sencillo la posible pérdida de prestaciones en el proceso de discretización.

Las prestaciones en bucle cerrado son comparadas no sólo en los instantes de muestreo, sino también en la respuesta *intermuestreo*, utilizando el comando `sdlsim`, para detectar, si las hubiera, posibles oscilaciones ocultas, por ejemplo. Otro caso de estudio detallando cómo utilizar `sdlsim` aparece en el vídeo [390].

Otro ejemplo de discretización de reguladores (en concreto, un control por modelo interno IMC) se aborda en el vídeo [418].

[388: drefin] Conclusiones/resumen sobre discretización de reguladores continuos

Materiales: [DiscRegContConclusiones.pdf]

*** 19:44


[YouTube ▶]

En este vídeo se recapitulan las ideas básicas sobre discretización de reguladores continuos vistas en detalle en otros vídeos de la colección: hay muchas opciones de discretización, muchas reglas de selección del período T ... a período pequeño todas las opciones funcionan bien, a período grande, la mayoría funciona mal. A veces puede sintonizarse un regulador por prueba y error, hay escenarios multifrecuencia... en resumen, todo es muy orientativo, y algunas de las limitaciones no son tales si se sintoniza un regulador discreto directamente (y no hay ruidos de proceso o medida de alta frecuencia que requieran precauciones especiales).

11.6.2 Caso de estudio, simulación sampled-data

[389: sdlsim1]

Caso estudio simulación intermuestreo (sampled-data) con
sdlsim (1): Planteamiento del problema


13:42

Materiales:
[[CÓD.:](#) SimulacionIntermuestreoBucleControlSDLSIM.mlx]
[[PDF](#)]
[[YouTube ▶](#)]

En este vídeo se discute la simulación de un bucle de control discreto, ante posibles problemas debidos a un período de muestreo excesivamente grande. En efecto, el proceso tiene modos de vibración poco amortiguados a frecuencias más allá de la frecuencia de Nyquist. La discretización ZOH no los tiene, claro. También se plantea un filtro anti-alias continuo que corte a la frecuencia de Nyquist.


En este vídeo se simulan las respuestas en bucle abierto y en bucle cerrado (mediante el comando `connect`), en tiempo discreto, con/sin ese filtro antialias (se observa distorsión, quizás poco aceptable). Se simula ante escalón en referencia y ante escalón en perturbación a la entrada del proceso.

Todo sería mejor con un período de muestreo más pequeño, pero se considera que no es factible en la aplicación en estudio.

Las conclusiones de las consideraciones en este vídeo es que se debería visualizar la respuesta intermuestreo para tener una idea más clara del funcionamiento del bucle. Esto se realiza en el vídeo [390], continuación de éste.

[390: sdlsim2]

Caso estudio simulación intermuestreo (sampled-data) con
sdlsim (2): simulación y discusión resultados


22:54

Materiales:
[[CÓD.:](#) SimulacionIntermuestreoBucleControlSDLSIM.mlx]
[[PDF](#)]
[[YouTube ▶](#)]

Este video es continuación del [389], donde se planteaba una simulación en tiempo discreto de un controlador de un proceso continuo (sistema muestreado). Los dos primeros minutos resumen las conclusiones del mencionado vídeo previo.

En este vídeo se plantea el diagrama de bloques (lft) necesario para el comando `sdlsim`: una interconexión linear-fractional-transformation entre un sistema en tiempo continuo y otro sistema en tiempo discreto, con muestreador y retenedor de orden cero entre ellos, generalizando a la simulación de sistemas híbridos el concepto de planta generalizada del vídeo [639].

También se discute cómo se incorporaría un filtro *antialias* analógico en el diagrama de bloques, caso de considerarse oportuno.


Se simula el bucle de control ante escalón en referencia, observandose oscilaciones ocultas debidas a una resonancia del proceso controlado más allá de la frecuencia de Nyquist. También se simula ante escalón en perturbación de entrada, y ante ondas senoidales en dicha perturbación de entrada. En concreto, si la perturbación de entrada tiene componentes más allá de la frecuencia de Nyquist, se observa, como era de esperar, un fenómeno ‘alias’ que el antialias analógico contribuye a mitigar (a costa de deformar bastante la respuesta sin él). Por culpa de ese fenómeno, el controlador en vez de cancelar una perturbación de alta frecuencia, introduce otra perturbación espúrea a baja frecuencia.

Capítulo 12

Control por modelo interno (IMC)

12.1 Idea general y metodología para plantas estables

[391: imc1] Control por modelo interno (IMC): diagrama de bloques e ideas básicas

***  09:38

Materiales: [IMC1dbloques.pdf]


[YouTube ►]

En este vídeo se presenta la idea básica del control por modelo interno: en una planta estable, lo único que necesita ser *realimentado* es la diferencia entre la *simulación* del modelo (que puede hacer *internamente* el regulador) y las observaciones.

Se presenta el diagrama de bloques que lleva a cabo esa idea, se introduce el parámetro de Youla Q (aunque Youla sí que considera la extensión a plantas inestables que en este material introductorio no se aborda, por simplicidad; se remite al lector interesado al vídeo [420]), y se detallan las funciones de transferencia en bucle cerrado entre las entradas exógenas (referencia y perturbaciones) y las variables clave en control (las variables controladas y la propia acción de control –variable manipulada–).

Las funciones de transferencia son lineales en Q (bueno, de forma rigurosa habría que decir que son *afines* en Q) y ello permite metodologías relativamente simples de diseño en la práctica (objetivo de otros materiales) y desarrollos teóricos varios, que tampoco se abordan aquí. La idea básica es que $Q \approx G^{-1}$, que daría lugar al bucle cerrado *ideal* $y = r$ y con cancelación total de perturbaciones (pero es imposible porque G^{-1} suele ser no realizable (ganancia a alta frecuencia tendiendo a ∞ , lo cual requiere un análisis más cuidadoso abordado en los vídeos [393] y [397]).

[392: imcQK] Control por modelo interno (IMC): equivalencia con controlador convencional

***  10:41

Materiales: [IMC2equivQK.pdf]

[YouTube ►]

Este vídeo discute la equivalencia entre un control por modelo interno (IMC) de parámetro Q , $u = Q(r - (y - Gu))$, cuyo diagrama de bloques e ideas básicas se detallan en el vídeo [391], y un control convencional de 1 grado de libertad $u = K(r - y)$.

Básicamente, es fácil de demostrar que dado Q , el regulador equivalente es $K = \frac{Q}{1-GQ}$ y, *viceversa*, dado K , el control IMC equivalente es $Q = \frac{K}{1+GK}$. El vídeo presenta la derivación de dichas expresiones bien mediante manipulaciones *gráficas* sobre el diagrama de bloques o, equivalentemente, mediante manipulaciones *algebraicas* sobre las ecuaciones de los bucles cerrados.

De las expresiones anteriores, se deduce el resultado teórico fundamental del IMC (que se puede extender al caso inestable Youla-Kucera, más complejo): si G es estable, un

regulador K estabiliza a G sí y sólo sí $Q = \frac{K}{1+GK}$ es estable, esto es, existe una biyección entre las funciones de transferencia estables Q y los reguladores K que estabilizan a G .

Esta equivalencia motiva dos posibles usos del IMC: (a) el *implícito* donde se usa Q para diseñar un regulador por la simplicidad de sus expresiones, pero luego se implementa K en la aplicación concreta, y (b) el *explícito* donde se simulan Q y G por separado siguiendo el diagrama de bloques original del IMC. En el caso lineal sin retardo, los dos enfoques son equivalentes; con retardo y no-linealidades el método *implícito* no es factible dado que K no sería una función de transferencia lineal de orden finito... el caso *explícito* con retardo se denomina *predictor de Smith*.

La parte final del vídeo demuestra que si $\text{dcgain}(G*Q)=1$, entonces el IMC tiene error de posición cero y, consecuentemente, el regulador K equivalente tiene acción integral.

Todas las ideas teóricas son aplicables también en tiempo discreto (con $G(z)$, $Q(z)$, etc.). Ejemplos Matlab de diseño de reguladores IMC en tiempo discreto (control por computador) se aborda en el vídeo [418] (discretización de diseños continuos) y el vídeo [419] (diseño directo en transformada \mathcal{Z}).

[393: imcfm] Control por modelo interno: metodología en procesos estables y de fase mínima

Materiales: [IMC3faseminima.pdf]

*** 10:37

[YouTube ►]

Este vídeo detalla cómo diseñar el parámetro Q de un control por modelo interno, cuyo diagrama de bloques (detallado en vídeo [391]) y funciones de transferencia de bucle cerrado se revisan en el primer minuto de vídeo.

Dado que la idea básica del IMC es $Q \approx G^{-1}$, se comentan los problemas de realizabilidad, inestabilidad si G es de fase no mínima, y de anticipación si G contiene retardo. No obstante, este vídeo aborda el caso de fase mínima y sin retardo, con lo que sólo se discutirán las cuestiones para que Q sea realizable.

La opción más sencilla es hacer $Q = G^{-1}M$ de modo que $y = Mr + (1 - M)(Gd_u + d_y)$. A la función de transferencia M se le denomina *modelo de referencia*. Si se escoge $M = 1/(\tau_M s + 1)^\nu$ siendo τ_M y ν los parámetros de diseño, se propone elegir un ν suficientemente grande para cancelar la subida de la ganancia a altas frecuencias de G^{-1} ($\nu \geq$ grado relativo de G) y un τ_M suficientemente pequeño para obtener la rapidez deseada de bucle cerrado (pero sin “pasarse” de rápido, para evitar saturación y amplificación excesiva de alta frecuencia en u).

La parte final del vídeo presenta un ejemplo concreto de diseño IMC para el proceso $G(s) = 4/(s + 1)$ con actuador saturando a ± 1 . Se escoge $M = 1/(0.25s + 1)$, y la simulación temporal de salida y acción de control resulta en principio satisfactoria. Dado que el IMC en procesos lineales es equivalente a un regulador convencional $K(s) = Q/(1 - QG)$ (detalles en vídeo [392]), se comprueba en este caso que resulta $K = \frac{M}{G(1-M)} = 1 + 1/s$, que tiene la estructura de un regulador PI. Por tanto, podría implementarse *explícitamente* Q y G o bien utilizarse el PI (sería un uso *implícito* del IMC para diseñar controladores convencionales).

Ejemplos adicionales de diseño se presentan en el vídeo [394].

[394: imcm11] Control por modelo interno de procesos estables y de fase mínima: ejemplo Matlab

Materiales: [[CÓD.: IMCexamples1.mlx](#)] [[PDF](#)]

*** 10:29

[YouTube ►]

En este vídeo se presenta el código Matlab que diseña y simula reguladores por modelo interno para el caso de procesos estables de fase mínima (sin retardo ni polos en el semiplano derecho). La base teórica se presenta en el vídeo [393].

En concreto, se diseñará para el proceso $4/(s+1)$ (ya visto en el vídeo referido), el proceso $5/(s+1)^2$, y el proceso $5 \cdot (0.15s+1)/(s+1)^3$. Se evaluarán distintas opciones en la selección del orden y constante de tiempo del modelo de referencia. Las soluciones deberán tener una respuesta temporal de salida y acción de control satisfactorias, así como respetar límites de saturación de actuador y evitar amplificar en exceso el ruido de alta frecuencia sobre la acción de control. Para procesos de orden bajo, el regulador IMC podría ser reinterpretado como un PID (vídeo [403]).

Otro ejemplo de diferentes alternativas de diseño para un proceso de segundo orden donde se simula la respuesta ante ruidos se detalla en los vídeos [395] y [396].

También, si el proceso a controlar es suficientemente rápido, puede proponerse un $Q = 1/\text{ganancia}$ constante, que no cambia el tiempo de establecimiento del proceso, ver vídeo [414], aunque lo usual es querer “acelerarlo un poquito”.

[395: imc4ca1]

IMC de proceso de segundo orden: ejemplo Matlab alternativas
diseño (I)

Materiales: [[CÓD.](#): EjemploIMCRUIDO.mlx] [[PDF](#)]

** 09:06

[YouTube ▶]

Este vídeo plantea el diseño de controladores IMC con un tiempo de establecimiento aproximado de 1.2 segundos para el proceso $G(s) = \frac{5}{(s+1)(0.15s+1)}$, como ejemplo de la metodología detallada en el vídeo [393].

En concreto, dado que con la fórmula de primer orden la constante de tiempo dominante es $\tau_c = t_{est}/4 = 0.3$, se discute el cálculo del parámetro Q del IMC y se presenta la respuesta ante escalón de los modelos de referencia: $\frac{1}{0.3s+1}$, $\frac{1}{(0.3s+1)^2}$, $\frac{1}{(0.3s+1)(0.03s+1)}$, $\frac{1}{(0.2s+1)^2}$ y el IMC “simplificado” donde el polo en $(s+1)$ se transforma en $(0.3s+1)$ mediante $Q = \frac{1}{5} \frac{(s+1)}{(0.3s+1)}$ correspondiendo implícitamente a un modelo de referencia $\frac{1}{(0.3s+1)(0.15s+1)}$.

Este vídeo discute únicamente el parámetro Q y la respuesta escalón de referencia a salida, conceptos éstos muy sencillos. Cuestiones más elaboradas, como la sensibilidad a ruido de medida, perturbaciones en la entrada y análisis en tiempo y en frecuencia se discuten en el vídeo [396], continuación de éste.

[396: imc4ca2]

IMC de proceso de segundo orden: alternativas diseño (2),
sensibilidad a ruido de medida y de proceso

Materiales: [[CÓD.](#): EjemploIMCRUIDO.mlx] [[PDF](#)]

**** 14:14

[YouTube ▶]

Este vídeo plantea el diseño de controladores con un tiempo de establecimiento aproximado de 1.2 segundos para el proceso $G(s) = \frac{5}{(s+1)(0.15s+1)}$.

En concreto, dado que con la fórmula de primer orden la constante de tiempo dominante es $\tau_c = t_{est}/4 = 0.3$, se comparan los modelos de referencia: $\frac{1}{(0.3s+1)^2}$, $\frac{1}{(0.3s+1)(0.03s+1)}$, $\frac{1}{(0.2s+1)^2}$ y el IMC “simplificado” donde el polo en $(s+1)$ se transforma en $(0.3s+1)$ correspondiendo a un modelo de referencia $\frac{1}{(0.3s+1)(0.15s+1)}$.

Una discusión un poco más amplia de cómo resulta el parámetro Q y la respuesta escalón de referencia a salida sin ruido se discute en el vídeo [395], resumida aquí en el primer minuto y medio; de todos modos, los conceptos son fáciles y la visualización de ese vídeo podría considerarse opcional.

En este vídeo se comparan la sensibilidad a ruido de medida, y a perturbaciones en la entrada mediante un análisis en tiempo (simulación ante ruido de medida) y en frecuencia (la introducción a la importancia del análisis en frecuencia de los bucles cerrados se discute en el vídeo [367]) de todos esos diseños.

El diseño de reguladores IMC en tiempo discreto (control por computador) se aborda en el vídeo [418] (discretización de diseños continuos) y el vídeo [419] (diseño directo en transformada Z).

Nota: Existe una generalización del IMC sencilla, a lo que se denomina control con 2 grados de libertad (procesado de referencia diferente al de medidas, $u = K_r r + K_y y$, en vez de $u = K(r - y)$). Esta generalización se discute en el vídeo [imc2gl(09:19)].

12.1.1 IMC en sistemas de fase no mínima

[397: imcfnm]

Control por modelo interno (IMC): caso fase no mínima (ceros semiplano derecho y retardo)

11:56

Materiales:

[IMC4faseNOminima.pdf]

[YouTube ▶]

En este vídeo se generaliza la metodología del control por modelo interno (vídeo [391]) al caso de procesos estables con ceros en el semiplano derecho y retardos (fase no mínima). El caso de fase mínima se discute en el vídeo [393].

Básicamente, el modelo de referencia debe contener el retardo y los ceros en el semiplano derecho (de otro modo $Q = G^{-1}M$ sería no causal e inestable, por lo que es necesario cancelar dichos elementos problemáticos de G^{-1} y la única forma es que estén en M . Esto es necesario para tener “estabilidad interna”, esto es, que todas las funciones de transferencia de bucle cerrado descritas en el vídeo [391] sean estables. La estabilidad interna de interconexiones de sistemas en representación interna se analiza formalmente en el vídeo [286], pero es más complejo y su visualización no es necesaria para comprender los conceptos aquí discutidos.

Dado que se van a cancelar, no es necesario incluir los elementos de fase no mínima en las manipulaciones para obtener Q : si se factoriza el proceso $G = G^+ G_0$ donde G^+ contiene los elementos de fase no mínima, se propone $Q = G_0^{-1} \frac{1}{(\tau_M s + 1)^\nu}$. En el vídeo se justifica dicha elección, y se presenta un ejemplo numérico de diseño para la planta $G(s) = 5 \frac{-0.15s+1}{(s+1)^2} e^{-0.1s}$, factorizándola como $G_0 = 5 \frac{0.15s+1}{(s+1)^2}$ y $G^+ = \frac{-0.15s+1}{0.15s+1} e^{-0.1s}$, de modo que G^+ es un filtro pasa-todo que, aproximadamente via la argumentación de Padé, puede interpretarse como *grosso modo* que nuestro objetivo es tener un retraso de ≈ 0.4 segundos en la respuesta respecto al modelo de referencia ideal sin retardo.

En procesos con retardo no se puede obtener un regulador como función de transferencia de orden finito usando $K = Q/(1 - QG)$, según explicado en el vídeo [392], dado que G hace que aparezcan *retardos internos*; de hecho, el tipo de sistema dinámico resultante es la base de lo que se denomina “predictor de Smith” en literatura. Se remite al lector a los ejemplos adicionales en Matlab discutidos en los vídeos [398] y [399]. La generalización a plantas inestables se presenta en el vídeo [420].

[398: imcm12]

Control por modelo interno (IMC) con cero de fase no mínima: ejemplo Matlab

08:28

Materiales:

[Cód.: IMCexamples2.mlx]
[PDF]

[YouTube ▶]

En este vídeo se discute el control por modelo interno del proceso $G(s) = 5 \frac{-0.5s+1}{(s+1)^2}$, usando las ideas teóricas discutidas en el vídeo [397].

El modelo de referencia “ideal” es $M_0(s) = \frac{1}{(\tau_M s + 1)^2}$, con $\tau_M = 0.2$; este modelo se obtendría si $G(s)$ fuera de fase mínima, pero en este caso no puede obtenerse sin cancelaciones polo-cero inestables que resultan en bucles cerrados donde no se cumple la estabilidad interna. Se comparan los resultados en tiempo y frecuencia de factorizar la planta $G = G_0 G^+$ como:

- $G_0(s) = 5 \frac{0.5s+1}{(s+1)^2}$, $G^+ = \frac{-0.5s+1}{0.5s+1}$
- $G_0(s) = 5 \frac{1}{(s+1)^2}$, $G^+ = -0.5s + 1$

Desde un punto de vista formal, ambas factorizaciones son válidas. La segunda de ellas da un comportamiento más rápido, pero el efecto de fase no mínima inicial (en dirección contraria a la referencia) es de un 70% comparado con sólo un 25% de la primera. En general, no es recomendable controlar procesos de fase no mínima con una rapidez excesivamente grande porque se intensifica el efecto y se reduce la robustez (tolerancia a error de modelado). El usar $G^+ = \frac{-0.5s+1}{0.5s+1}$ es un mecanismo de “salvaguarda” dado que si $\tau_M < 0.5$ el polo en $s = -2$ no cancelado de G^+ que aparece debido a $(0.5s + 1)$ sería el dominante.

[399: imcml3] Control por modelo interno con retardo y ceros de fase no mínima: ejemplo Matlab

Materiales: [[CÓD.: IMCexamples2.mlx](#)] [[PDF](#)]

**** 05:16

[YouTube ▶]

Este vídeo presenta ejemplos adicionales de la metodología de control por modelo interno, para los procesos $G(s) = \frac{5}{s+1} \cdot e^{-1.5s}$, esto es, primer orden + retardo, y $G(s) = 5 * \frac{-0.35s+1}{(s+1)^2} \cdot e^{-0.5s}$ donde aparecen retardo y cero en semiplano derecho simultáneamente, complementando las ideas teóricas del vídeo [397] y el ejemplo del vídeo [398]. Ejemplos adicionales, y las expresiones de los reguladores IMC de estos procesos de bajo orden como PIDs, se pueden visualizar en el vídeo [403].

[400: imcnova] Control por modelo interno: contraejemplos (cuando NO funciona bien)

Materiales: [[CÓD.: IMCexamples3NoFunciona.mlx](#)]

*** 08:00

[YouTube ▶]

Este vídeo presenta ejemplos de código donde **no** funciona el control por modelo interno. Primero, se aborda el caso de un proceso inestable. Si se ejecuta el código de los ejemplos del vídeo [394], se alcanza el modelo deseado entre referencia y salida pero hay una cancelación polo-cero inestable entre regulador y planta que hace que la respuesta ante perturbaciones a la entrada del proceso tienda a infinito. Esto no es ninguna sorpresa, dado que está previsto por la teoría. Esa inestabilidad ante perturbación a la entrada de $(1 - M) \cdot G$ no es un problema “numérico” de simulación: es intrínseco a la filosofía de diseño y simulando `minreal((1-M)*G)` tampoco funciona, ni lo haría en un experimento real.

El segundo ejemplo es un proceso $G(s) = 10/(s + 0.004)/(0.05s + 1)$. El proceso es estable, por lo que la teoría del IMC no predice ningún problema *a priori*. No obstante, la cancelación polo-cero casi en el límite de la estabilidad produce una respuesta ante perturbaciones a la entrada de la planta que tarda 600 veces más que la respuesta ante referencia. Ello no suele ser aceptable en la mayor parte de las aplicaciones.

El límite de la idea de este segundo ejemplo es un proceso con integrador puro (*inestabilidad marginal* $1/s$) en el cual el IMC tampoco funcionará. Un ejemplo de ese tipo aparece en el vídeo [413]... en dicho vídeo se usa un “S-IMC” que es una modificación que, pese a su nombre con el acrónimo IMC, realmente abandona la filosofía IMC en procesos con polos muy lentos (teoría en vídeo [404]).

Un caso de IMC en tiempo discreto y análisis de este tipo de problemas debidos a la cancelación se aborda en el vídeo [287], aunque su visualización se aconseja sólo si ya te son familiares los conceptos de controlabilidad y observabilidad en representación interna.

Aparte de estos casos “extremos”, el IMC (y, realmente, cualquier controlador) demasiado “enérgico” tendrá problemas de amplificación de ruido o saturación que pueden desaconsejar un diseño “teóricamente correcto” en determinada aplicación, ver vídeo [396].

12.2 Sistemas con retardo: predictor de Smith

[401: imcsmith]

Sistemas con retardo: predictor de Smith (derivación a partir de IMC)

Materiales: [IMC5RetardoSmith.pdf]

09:31

[\[YouTube ►\]](#)

Este vídeo discute la derivación de la estructura de control que se conoce como *Predictor de Smith* a partir de la formulación del control por modelo interno (vídeo [391]); esta estructura se aplica para controlar procesos con retardo $G = e^{-\theta s} \cdot \tilde{G}$ a partir de controladores \tilde{K} que funcionan correctamente con el modelo nominal (desprovisto del retardo) \tilde{G} .

Realmente es un desarrollo casi paralelo al del vídeo [397], de modo que no se presenta aquí ninguna herramienta que permita hacer nada diferente a lo ya explicado en el referido vídeo: el predictor de Smith es un caso particular del IMC.

En el predictor bajo estudio aquí se propone escoger, como parámetro Q del IMC con el proceso G con retardo, el mismo parámetro Q que utilizaríamos para un control IMC del proceso \tilde{G} ; de hecho, este parámetro Q se deriva con $Q = \frac{\tilde{K}}{1 + \tilde{G} \cdot \tilde{K}}$ si ya se ha diseñado previamente un regulador \tilde{K} que funciona bien con \tilde{G} con cualquier técnica sea o no IMC (usando la equivalencia discutida en el vídeo [392]). Manipulaciones sencillas del IMC resultante con G dan lugar al citado predictor de Smith.

[402: smithml]

Predictor de Smith para control de sistemas con retardo: ejemplo Matlab

Materiales: [[CÓD.: smithexample.mlx](#)]

09:51

[\[YouTube ►\]](#)

Este vídeo presenta un ejemplo numérico de cálculo y simulación del predictor de Smith, cuya teoría basada en el control por modelo interno se aborda en el vídeo [401].

Como regulador nominal \tilde{K} , se usa un PID con filtro de ruido, sintonizado manualmente para tener una respuesta razonablemente satisfactoria para el proceso de segundo orden nominal $\tilde{G}(s) = 5/(s + 1)/(0.45s + 1)$. Cuando se introduce un retardo de 1.5 segundos en la planta, esto es, $G = e^{-1.5s} \tilde{G}$, el PID se hace inestable.

El predictor de Smith, $(1 - e^{-1.5s})\tilde{G}$, conectado con realimentación negativa con \tilde{K} , sí que consigue lo que pretendía la teoría: que la respuesta ante referencia sea igual a la nominal pero retrasada.

El vídeo analiza, además de la respuesta temporal ante referencia, la respuesta en frecuencia (cancelación de perturbaciones/ruido) ante perturbaciones de salida y de entrada.

La última parte del vídeo analiza qué ocurre si el retardo del predictor de Smith (modelo interno) no coincide con el retardo real de la planta. Se observa que cuanto más difieren, más se degradan las prestaciones, como era de esperar intuitivamente.

12.3 Diseño de PID's con IMC

[403: imcpid]

Diseño de PID's por medio de la metodología de modelo interno (IMC-PID)

Materiales: [[CÓD.: PIDviaIMC.mlx](#)] [[PDF](#)]

10:26

[\[YouTube ►\]](#)

Este vídeo plantea específicamente el diseño de PIDs para procesos estables de tipo $G(s) = \frac{1}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-\theta s}$, esto es, primer o segundo orden mas retraso, sin ceros. La obtención de esos modelos a partir de datos experimentales se aborda, por ejemplo, en los vídeos [154] y [748].

Básicamente, sin retardo, el IMC con modelo de referencia $M = \frac{1}{\tau_c s + 1}$ resulta en una función de transferencia que podría transformarse a un PID, usando las ideas sencillas expuestas en el vídeo [516] y, de hecho, no resulta muy diferente a los ejemplos del vídeo [394].

Con retardo y $M = \frac{1}{\tau_c s + 1} e^{-\theta s}$, el resultado es un regulador con retardos internos (como los vistos en el vídeo [399], o similar a los que motivan la metodología del predictor de Smith en el vídeo [401]). Para transformarlo en un PID, sin retardos internos, el término $1 - e^{-\theta s}$ se aproxima por su serie de Taylor, y resulta $1 - e^{-\theta s} \approx \theta s$.

Incorporando esa última aproximación, el vídeo presenta la estructura IMC-PID serie resultante:

$$PID = \underbrace{\frac{\tau_1}{k(\tau_c + \theta)}}_{\text{Proporcional}} \cdot \underbrace{\left(1 + \frac{1}{\tau_1 s}\right)}_{\text{Integral}} \cdot \underbrace{(\tau_2 s + 1)}_{\text{Derivada}}$$

El vídeo presenta un ejemplo numérico de diferentes diseños para $G(s) = \frac{1}{s+1} e^{-0.05s}$, viendo cómo cambian las distintas respuestas temporales de bucle cerrado conforme el parámetro de diseño τ_c se modifica. Si aceleramos mucho el bucle cerrado (por ejemplo, algunos autores sugieren limitarse a $\tau_c \leq \tau_1/4$) las limitaciones de la metodología son la lentitud de la respuesta ante perturbaciones de entrada y que se notan más las deficiencias de la aproximación del retardo.

Nota: como todo regulador integral, el IMC o IMC-PID resultante presenta problemas de windup ante saturaciones. Estas cuestiones son discutidas en el vídeo [417].

[404: simcteo]

PIDs con IMC para procesos lentos: modificación S-IMC
(Skogestad), teoría

Materiales: [[CÓD.: PIDviaIMCvsSIMC.pdf](#)]

11:00

[YouTube ▶]

Este vídeo presenta una breve revisión de la metodología S-IMC (Skogestad IMC) de sintonía de PIDs basada en modelo interno, para plantas lentas o integradores.

El IMC para plantas con polos cerca del límite de la estabilidad funciona “teóricamente” pero tiene inconvenientes en la respuesta ante perturbaciones a la entrada del proceso (detalles en vídeo [400]). Esos efectos ocurren también en la sintonía de PIDs basados en es IMC (discutida en el vídeo [403]). La modificación S-IMC propone cambiar el PID a:

$$PID = \underbrace{\frac{\tau_1}{k(\tau_c + \theta)}}_{\text{Proporcional}} \cdot \underbrace{\left(1 + \frac{1}{\min(\tau_1, 4(\tau_c + \theta))s}\right)}_{\text{Integral}} \cdot \underbrace{(\tau_2 s + 1)}_{\text{Derivada}}$$

Ello produce una mejor respuesta ante perturbaciones en la entrada porque el bucle ya no tiene cancelaciones lentas entre regulador y proceso controlado.

Este vídeo discute la justificación teórica de esta propuesta (ejemplos numéricos en simulación aparecen en el vídeo [405]).

La primera idea se basa en considerar, por sencillez, un caso límite donde el polo de la planta sea mucho más lento que los objetivos de control $\tau_1 \gg \tau_c$. Entonces la planta $G = k/(\tau_1 s + 1)$ se puede aproximar por $k/(\tau_1 s)$, y la propuesta de arriba consigue un par de polos (amortiguamiento crítico) a la mitad de la rapidez inicialmente planificada en τ_c pero al evitar la cancelación, la respuesta ante perturbaciones de entrada también desaparece con esa rapidez.

La parte final del vídeo discute qué ocurre con un proceso que no es exactamente un integrador: una situación intermedia donde el polo más “visible” en la respuesta ante

referencia se hace algo más lento que el de $(\tau_c s + 1)$ y el polo asociado a la respuesta ante perturbaciones en la entrada se hace algo más rápido que el de $(\tau_1 s + 1)$.

[405: [simcml](#)] [PIDs con IMC para procesos lentos: modificación S-IMC \(Skogestad\), ejemplo Matlab](#)

Materiales: [[CÓD.:](#) PIDviaIMCvsSIMC.mlx] [[PDF](#)]

*** 10:27

[YouTube ▶]

Este vídeo presenta una breve revisión de la metodología S-IMC (Skogestad IMC) de sintonía de PIDs basada en modelo interno, para plantas lentas o integradores. En efecto, el IMC para plantas con polos cerca del límite de la estabilidad funciona “teóricamente” pero tiene inconvenientes en la respuesta ante perturbaciones a la entrada del proceso (ver detalles sobre este tema en vídeo [400]). Esos efectos ocurren también en la sintonía de PIDs basados en esa idea (discutida en la parte final del vídeo [403]). La modificación S-IMC propone cambiar el PID a:

$$PID = \underbrace{\frac{\tau_1}{k(\tau_c + \theta)}}_{\text{Proporcional}} \cdot \underbrace{\left(1 + \frac{1}{\min(\tau_1, 4(\tau_c + \theta))s}\right)}_{\text{Integral}} \cdot \underbrace{(\tau_2 s + 1)}_{\text{Derivada}}$$

Ello produce una mejor respuesta ante perturbaciones en la entrada porque el bucle ya no tiene cancelaciones lentas entre regulador y proceso controlado.

El objetivo de este vídeo es comprobar mediante distintas simulaciones el efecto de esa modificación, comparando la respuesta con la del IMC-PID sin modificar. Otro ejemplo más sencillo de primer orden sin retardo se desarrolla en el vídeo [406].

El lector interesado puede consultar una justificación más detallada de la idea teórica subyacente (lugar de las raíces) en el vídeo [404].

[406: [imcpiord1](#)] [Diseño de PI mediante IMC para proceso de primer orden, y modificación S-IMC: ejemplo Matlab](#)

Materiales: [[CÓD.:](#) disseny1erordIMCPI.mlx] [[PDF](#)]

*** 06:40

[YouTube ▶]

En este vídeo se diseña un IMC para acelerar 9 veces la constante de tiempo de un proceso $G = 4/(s + 1)$ buscando un comportamiento ante referencia dado por $M_{ref} = 1/(s + 9)$. Para ello, se propone $Q = G^{-1}M_{ref}$. Se calcula $K = Q/(1 + GQ)$, y mediante **residue** se expresa como un PID en forma paralela.

La segunda parte del vídeo discute que, dado que se ha acelerado 9 veces la respuesta ante referencia, en muchos casos la respuesta ante perturbaciones de entrada (con polo en $s = -1$) resulta no satisfactoria. Por ello, se propone la modificación de Skogestad $\frac{\tau_1}{\kappa\tau_c} \cdot \left(1 + \frac{1}{\tau_I s}\right)$, con $\tau_I = \min(\tau_1, 4\tau_c)$ que en este caso resulta $4\tau_c$. Se simula la respuesta ante referencias y perturbaciones del nuevo diseño (recomendado) y se compara con el anterior (no recomendado). Se pueden ver más ejemplos de S-IMC en el vídeo [405], y justificación teórica en el vídeo [404].

Caso de estudio

[407: [imcce1](#)] [Ejemplo de diseño IMC \(2o ord+retraso\) y transformación a PID en forma paralela](#)

Materiales: [[CÓD.:](#) IMCyPIDDisseny.mlx] [[PDF](#)]

*** 13:16

[YouTube ▶]

Este fichero presenta el diseño IMC para un proceso de segundo orden + retardo. Plantea un modelo sin retardo M_+ (el modelo sería M_+e^{-ds}), un modelo de referencia M_{ref} y calcula $Q = M_+^{-1}M_{ref}$. Se presenta la respuesta ante escalón de $MQ = M_{ref}e^{-ds}$.

Con modelo de referencia de orden 2, Q sale realizable. Con modelo de referencia de orden 1, entonces Q no es realizable.

La segunda parte del vídeo discute la transformación del IMC a un PID paralelo $K_p + K_i/s + K_d s \cdot N/(s + N)$. Primero se aproxima el retardo, par obtener una FdT aproximada $K_{implicit}$ del IMC implícito sin retardos internos mediante realimentación positiva. Luego se descompone en fracciones simples y se identifican los residuos obtenidos con los coeficientes del PID en la forma requerida.

Si se hubiese escogido modelo de referencia de orden 1, el PID hubiese resultado no realizable y se necesitaría añadir “manualmente” un filtro de ruido de un cierto ancho de banda dependiente de la rapidez del proceso y de la calidad de los sensores. Esto se discute en la parte final del vídeo.

[408: imcce2s] Simulación con Simulink de tres posibles implementaciones de un IMC (explícito, implícito-sin ret.interno-, PID): diagrama de bloques

Materiales: [CÓD.: SimulinkIMC3imp.slx] [PDF]

*** 07:39

[YouTube ▶]

Este fichero detalla un modelo de Simulink donde aparece un diagrama de bloques de un proceso con saturación, ancho de banda limitado de actuador, ruidos de proceso y medida y perturbación escalón de entrada. Este proceso es “parecido” pero no igual al usado para diseñar un IMC en el vídeo [407].

Los diagrama de bloques que implementan el IMC explícitamente, el IMC implícito aproximado (esto es, aproximando el retardo para que sea una función de transferencia sin retardos internos) y el PID en forma paralela equivalente a dicho regulador implícito, ver detalles en el referido vídeo [407].

Se presentan sub-modelos simulando el proceso, los controladores, la selección del tipo de controlador que se prueba con el proceso y la interfaz de usuario (selección de referencia y gráficas de respuesta).

[409: imcce3s] Simulación con Simulink de tres posibles implementaciones de un IMC (explícito, implícito-sin ret.interno-, PID): resultados

Materiales: [CÓD.: SimulinkIMC3imp.slx]

*** 08:15

[YouTube ▶]

En este vídeo se simulan los controladores IMC explícito y IMC-PID resultantes del diseño del vídeo [407]. El diagrama de bloques del modelo Simulink ha sido descrito en el vídeo [408].

Los resultados confirman que el PID y el IMC producen respuestas casi indistinguibles (la aproximación del retardo es suficientemente exacta), cercanas al modelo ideal incluso con perturbaciones y pequeños errores de modelado, si la saturación sólo ocurre durante una fracción despreciable de la duración del transitorio.

La segunda parte del vídeo considera referencias más altas donde si la saturación ocupa gran parte del transitorio la respuesta difiere bastante de la “ideal”, y si la referencia supera la ganancia por el máximo de la amplitud de la variable manipulada, entonces (siempre se satura) se produce el fenómeno de wind-up debido a la presencia de acción integral y errores que no se anulan. Aunque Q no tenga un integrador, ese integrador está “oculto” y se ve en la transformación a IMC-PID. El windup se produce en ambas implementaciones. Por ello, sería necesario en la práctica incluir estrategias antiwindup que no se discuten en este vídeo por brevedad (ver vídeo [417]).

12.4 Ejercicios adicionales

[410: imcexau] Funciones para diseño/simulación de IMC/S-IMC (Matlab)
Materiales: [[CÓD.:](#) SIMCAux.zip] [[PDF](#)]

**** **07:49**
 [YouTube ▶]

En este vídeo se explica el código Matlab para calcular de forma automática el PID/S-IMC utilizando las tablas de Skogestad, así como código que los simula ante cambios escalón de referencia y de perturbación a la entrada del proceso controlado. Este código se utiliza como auxiliar en otros ejemplos.

[411: imcex1] Ejercicio: Análisis de opciones IMC/S-IMC para proceso de segundo orden (Matlab)
Materiales: [[CÓD.:](#) IMCEjerc1.mlx] [[PDF](#)]

*** **09:10**
 [YouTube ▶]

Este vídeo discute las posibles opciones IMC o PID/S-IMC para el proceso:

$$G(s) = \frac{2 \cdot (0.5s + 1)}{(12s + 1)(5.5s + 1)}$$

Primero se hace un análisis cualitativo de las opciones disponibles (tipo de diseño y especificaciones aconsejables). En la segunda parte del vídeo se simulan dichas opciones para comprobar el comportamiento conseguido.

*El código hace uso de las funciones auxiliares discutidas en el vídeo [410].

Para estudiantes interesados (opcional): el diseño de reguladores IMC en tiempo discreto (control por computador) se aborda en el vídeo [418] (discretización de diseños continuos) y el vídeo [419] (diseño directo en transformada \mathcal{Z}).

[412: imcex2] Ejercicio: Análisis de opciones IMC/S-IMC para proceso de 2o orden + retardo (Matlab)
Materiales: [[CÓD.:](#) IMCEjerc2.mlx] [[PDF](#)]

*** **06:30**
 [YouTube ▶]

Este vídeo discute las posibles opciones IMC o PID/S-IMC para el proceso:

$$G(s) = \frac{1}{(10s + 1)^2} \cdot e^{-4.5s}$$

Primero se hace un análisis cualitativo de las opciones disponibles (tipo de diseño y especificaciones aconsejables). En la segunda parte del vídeo se simulan dichas opciones para comprobar el comportamiento conseguido.

*El código hace uso de las funciones auxiliares discutidas en el vídeo [410].

[413: imcex3] Ejercicio: Análisis de opciones IMC/S-IMC para proceso integrador+retardo (Matlab)
Materiales: [[CÓD.:](#) IMCEjerc3.mlx] [[PDF](#)]

*** **05:01**
 [YouTube ▶]


Este vídeo discute las posibles opciones IMC o PID/S-IMC para el proceso:

$$G(s) = \frac{1.5}{s \cdot (3.5s + 1)} \cdot e^{-16s}$$

Primero se hace un análisis cualitativo de las opciones disponibles (tipo de diseño y especificaciones aconsejables). En la segunda parte del vídeo se simulan dichas opciones para comprobar el comportamiento conseguido.

En este caso, al ser el proceso (marginalmente) inestable, el IMC no debería ser usado, según la teoría. La parte final del vídeo comprueba dicho mal funcionamiento y, también, se recalca que el S-IMC no es la *única* opción de sintonizar PIDs. Se compara la respuesta del PID/S-IMC con, por ejemplo, el resultado de utilizar **pidtune** con la opción 'DesignFocus' establecida a 'disturbance-rejection'. Esta idea es *importante*: el IMC o PID/S-IMC no tiene por qué ser ni la "única" opción ni la "mejor" en un caso genérico.

*El código hace uso de las funciones auxiliares discutidas en el vídeo [410].

[414: imcQcte]	Control de procesos rápidos: IMC con Q constante (1/ganancia)	***  09:20
Materiales: [Cód.: IMCQct.mlx] [PDF]		[YouTube ►]


*[LINK TO ENGLISH VERSION](#)

Este vídeo discute el control IMC de un proceso "rápido" donde su tiempo de establecimiento en bucle abierto sea ya rápido, para una aplicación concreta, y no desee acelerarlo en bucle cerrado para, por ejemplo, evitar sobredimensionar los actuadores innecesariamente.

En resumen: polos de bucle cerrado en el mismo sitio que bucle abierto se considera aceptable.

En este caso, el IMC resulta fácil, se trata simplemente de proponer $Q=1/\text{dcgain}(G)$. Al transformarlo a un regulador convencional (IMC implícito) con $K = Q/(1 - GK)$ resulta en un controlador con acción integral en bucle cerrado, como la teoría predice, de modo que ante perturbaciones de entrada, salida y ante cambios de referencia, presenta una dinámica que hace el error cero con unos polos de bucle cerrado en el mismo sitio que los polos de bucle abierto, que era nuestro objetivo.

Se representa la respuesta temporal tanto ante cambios de referencia como ante perturbación a la entrada (ruido de proceso).


[415: imcex4]	Ejercicio: Análisis de opciones IMC/S-IMC para proceso 2o orden+retardo	X. Blasco, J.M. Herrero, C. Ramos, A. Sala (UPV)	*** 
Materiales: [Cód.: IMCEjerc4.mlx] [PDF]			

Este ejemplo Matlab discute las posibles opciones IMC o PID/S-IMC para el proceso:

$$G(s) = \frac{2.5}{(10s + 1)(6s + 1)} \cdot e^{-7s}$$

Primero se hace un análisis cualitativo de las opciones disponibles (tipo de diseño y especificaciones aconsejables). Luego se simulan dichas opciones para comprobar el comportamiento conseguido. Como el desarrollo es parecido a otros ejercicios, no se ha considerado necesario incluir la explicación en vídeo.

*El código hace uso de las funciones auxiliares discutidas en el vídeo [410].

[416: imcex5]	Ejercicio: Análisis de opciones IMC/S-IMC para proceso 2o orden+retardo		*** 
Materiales: [Cód.: IMCEjerc5.mlx] [PDF]			

Este ejemplo Matlab discute las posibles opciones IMC o PID/S-IMC para el proceso:

$$G(s) = \frac{3}{(22s + 1)(6s + 1)} \cdot e^{-2s}$$

El análisis es paralelo a otros ejercicios; no se ha considerado necesario incluir la explicación en vídeo.

*El código hace uso de las funciones auxiliares discutidas en el vídeo [410].

12.4.1 Antiwindup

[417: imcantiwind]
10:11

IMC ante saturación: windup y modificación antiwindup.
Comparación clamp-antiwindup en PID equivalente.

**** 

Materiales: [[CÓD.:](#) testAWSimpleS.zip] [[PDF](#)]

[YouTube ►]

Los IMC donde GQ tenga ganancia 1 (que es lo usual) incorporan *implícitamente* un integrador (teoría en parte final de vídeo [392]). Por ello, el efecto *windup* donde dicho integrador acumula error indebidamente cuando el actuador está saturado se produce tanto en la implementación *explícita* como *implícita*. En este video, se ilustra con un IMC para el proceso $4/(s + 1)$ y en el PI que resulta equivalente a dicho IMC (detalles del diseño en vídeo [406]).


Una sencilla forma de evitar el *windup* en IMC explícito es alimentar al modelo interno con la acción saturada, de modo que el modelo interno recibe la misma acción que el proceso real y, por tanto, no se “rompe” la idea básica: la saturación es parte del modelo. En efecto, el *windup* se evita, pero los antiwindup clásicos sobre el PI equivalente (sin saturación) tienen una *mejor respuesta* que el IMC; en efecto, el IMC – que tiene un comportamiento ante referencia dado por $G \cdot sat \cdot Q$ – disminuye el control demasiado pronto, debido a que Q ha sido calculado pensando en las propiedades *bucle abierto* de $G \cdot Q$ (que sube más rápido). El regulador PI, sin embargo, al realimentar el error se da cuenta de que no se ha hecho tan pequeño como la simulación sin saturación predeciría y continua produciendo una acción de control grande durante más tiempo.

El antiwindup en reguladores PID se aborda en el vídeo [521], y en reguladores genéricos en representación interna, en el vídeo [673].

12.4.2 Diseño IMC para implementación por computador (tiempo discreto)

[418: imcdt1]

Control IMC en tiempo discreto (1): discretización de IMC
continuo

****  15:38

Materiales: [[CÓD.:](#) testIMCdiscretoCT.mlx] [[PDF](#)]

[YouTube ►]

En este video se discute un diseño IMC continuo para un proceso $G(s)$ de 2o orden poco amortiguado, según discutido en otros materiales (teoría en [391], [392]; ejemplo Matlab en [394] y otros).

Una vez ese diseño está finalizado, se proponen tres discretizaciones:

- Discretizar el regulador $K(s)$ por ‘tustin’, siendo $K(s) = Q/(1 - QG)$ el regulador con IMC ‘implícito’.
- Discretizar el regulador $K_2(s)$ por ‘tustin’, siendo $K(s) = Q/(1 - QG \cdot (1 - \frac{T_s}{2}s))$, incorporando el efecto de un muestreo como medio período de retardo (fase no mínima).

- Discretizar $Q(s)$ por tustin, y formar $K_d(z) = Q_d/(1 - Q_d G_d)$, siendo G_d la discretización ZOH exacta de $G(s)$.

Se comprueba que, para período de muestreo T_s pequeño, la segunda y tercera opciones dan un resultado muy parecido al diseño continuo de partida (y la primera tampoco está tan mal). Obviamente, los reguladores IMC tienen un problema con perturbaciones de entrada o condiciones iniciales no nulas (no se amortigua la respuesta a ellas) que, dependiendo de los requisitos de la aplicación podría hacer necesaria otra metodología de diseño.

El diseño directo en tiempo discreto para el mismo proceso se aborda en el vídeo [419], continuación de éste.

[419: imcdt2] **Control IMC en tiempo discreto (2): diseño $Q(z)$ puramente discreto**

Materiales: [[CÓD.:](#) testIMCdiscretoDirecto.mlx] [[PDF](#)]

**** 16:55

[YouTube ▶]

En este video se discute un diseño IMC para un proceso $G(s)$ de segundo orden poco amortiguado. Primero se discute el diseño en tiempo continuo (idéntico al del vídeo [418]) y a continuación se plantea el objetivo de control: obtener unas prestaciones similares en tiempo discreto.

La primera opción sería discretizar adecuadamente el diseño continuo; esto se ha hecho en el vídeo [418], y para períodos pequeños podría ser una solución perfectamente válida. .

En el presente material se operará directamente en tiempo discreto: se discretiza el modelo de referencia y, a partir del modelo discreto ZOH del proceso se obtiene el parámetro $Q(z)$ del IMC. Se propone como razonable un período de muestreo $T_s = 0.25$ segundos, comparando con otras opciones (en discretización de diseño continuo se optó por recomendar $T_s = 0.1$, suponiendo que no existen perturbaciones de frecuencia mayor que requieran aumentar la frecuencia de muestreo).

Una vez elegido T_s , se argumenta que conviene observar los ceros de muestreo en el modelo de la planta a controlar... si están cerca del círculo unidad y se cancelan entonces se producirá un polo cerca del círculo unidad en el controlador, que dará oscilaciones poco recomendables (la idea es similar a la discutida en el vídeo [400]) . Por eso se construye un factor de corrección sobre el modelo de referencia para mantener los ceros de muestreo de la planta "original" a controlar y no incluir los ceros de muestreo del modelo de referencia del diseño continuo.

El vídeo [287] presenta un análisis más detallado de los inconvenientes por cancelar polos y ceros cerca del círculo unidad, y su relación con los conceptos de controlabilidad y observabilidad en espacio de estados.

Tras el diseño, se simula, en tiempo discreto (no se hace simulación intermuestreo, que requeriría ejecutar el comando `sdlsim`, ver vídeo [390]) el diseño, tanto ante escalones de referencia, como de perturbación a la entrada (que en IMC puede dar problemas). También se representa la respuesta en frecuencia de bucle cerrado y se prueban varios períodos de muestreo.

12.5 Caso general inestable: parametrización de Youla-Kucera

[420: YQparam] **Parametrización de Youla-Kucera (generalización de IMC): caso general (planta inestable)**

13:41

Materiales: [YKdbloques.pdf]

[YouTube ▶]

Este vídeo discute la generalización del control por modelo interno (vídeo [391]) a la denominada parametrización de Youla-Kucera que incluye el caso de plantas inestables. Parte de un controlador K_0 preexistente que se supone que estabiliza una planta G . A partir de la factorización de $K_0 = VX^{-1}$ y de $G = ND^{-1}$ en “numerador” y “denominador”, se plantea un diagrama de bloques que añade el parámetro Q y el modelo interno de la planta al diagrama original. Con ello se obtienen las ecuaciones de bucle cerrado y se enuncia (omitiendo demostración por brevedad y por requerir detalles técnicos sobre doubly-coprime factorisations) el resultado principal que dice que $K(Q) = (V + DQ)(X + NQ)^{-1}$ también estabiliza a la planta ND^{-1} si Q es estable y, viceversa, si K_1 estabiliza a la planta, existe un Q_1 estable de modo que puede escribirse como $K_1 = K(Q_1)$.

La última parte del vídeo explica que usando $K_0 = 0 \cdot 1^{-1}$ y $G = G \cdot 1^{-1}$ las fórmulas de $K(Q)$ y el diagrama de bloques se reducen al IMC para plantas estables discutidas en el vídeo [392].

Esta parametrización tiene mucha importancia teórica. Sin embargo, dado que las funciones de transferencia de bucle cerrado son más complicadas que el IMC estable (en ese caso sencillo, se tiene $T = GQ$), por incluir el controlador inicial, la importancia práctica es menor porque no da lugar con facilidad a reglas “sencillas” de diseño.

Parte III


Control Multivariable

Capítulo 13


Control de procesos complejos: etapas iniciales

Introducción: El control de procesos complejos no es “fácil” pese a que haya técnicas de control multivariable que diseñan controladores con “*una línea de Matlab*”... hay muchas decisiones *clave* a tomar antes, de gran importancia económica, algunas de las cuales se tratarán en este capítulo.

13.1 Problemas a resolver, selección de punto de operación

[421: cscmot]	Control de sistemas complejos (motivación): preguntas a responder antes de calcular controladores	**  08:19
	Materiales: [MotivacionPtOper.pdf]	[YouTube ►]

Este vídeo habla de “lo que no se suele hablar en cursos de control”: las muchas cuestiones previas antes de hacer “Matemáticas” con $y(s) = G(s)u(s) + H(s)d(s)$. Para llegar ahí hay que discutir qué se elige como variable controlada, qué como variable manipulada (y qué tecnologías y potencias de actuadores), qué variables se miden (incluyendo variables auxiliares), qué complejidad se le da al modelo y, muchas veces, estas decisiones vienen condicionadas por cuestiones de seguridad y tolerancia a fallos, de gran importancia en ciertas aplicaciones peligrosas (lo que ganas en 2 años lo pierdes en 2 segundos). La discusión sobre las distintas fases de un proyecto de control de sistemas complejos, y el papel en todo esto de la Ingeniería de Sistemas y Automática continua en el vídeo [422].

[422: csceta]	Control de sistemas complejos: etapas y papel de la Ingeniería de Sistemas y Automática	**  12:00
	Materiales: [MotivacionPtOper.pdf]	[YouTube ►]

Este vídeo habla de “lo que no se suele hablar en cursos de control”, continuando la discusión del vídeo [421]. Las muchas cuestiones previas antes de hacer “Matemáticas” con $y(s) = G(s)u(s) + H(s)d(s)$ acaban necesitando abordar un problema de control de un sistema complejo desde la fase de proyecto (proyectar sistemas “fáciles de controlar”) y luego ya, en producción, como etapa posterior a planificación estratégica, determinación de puntos de funcionamiento óptimos, selección de variables controladas, manipuladas y medidas, y determinación de qué especificaciones de control deben cumplir los bucles elegidos. Una pequeña ampliación de la discusión sobre estas etapas a seguir en control de sistemas complejos aparece en el vídeo [423].

Las herramientas de ingeniería de sistemas y automática (señales y sistemas) proveen de un punto de vista genérico interdisciplinar ($1/(3s+1)$) vale para controlar presión, temperatura

o velocidades) que resulta beneficioso en muchos casos, pero también están limitadas al no conocer aspectos clave sobre tecnologías específicas del proceso a ser controlado (tecnologías de actuación, de medición, límites de prestaciones, normativas de seguridad, química, física, fluidos,...): la ingeniería de procesos industriales no se reduce a dominar Matlab y Simulink.

[423: layers] Niveles de decisión en control de procesos (plantwide control)

Materiales: [PtOper.pdf]

*  04:11

[YouTube ►]

Este vídeo describe los niveles del control de sistemas complejos: las capas de 1.-planificación, 2.-optimización y 3.-control, donde cada capa da referencias a la siguiente.

Diferentes niveles de decisión (planificación de producción –plazo de días, semanas o meses–, optimización de punto de operación –plazo de horas–, control –plazo de minutos–). Idealmente podría resolverse con un único optimizador, pero no es así en la práctica dadas las diferentes filosofías requeridas para abordar cada problema. La ingeniería de control usualmente aborda la capa 3 y, en ocasiones, la 2.

Se recomienda visualizar el vídeo [424] para comprender mejor el significado de optimización de un punto de operación, así como el caso de estudio sobre climatización en los vídeos [425] y [426] para esbozar la aplicación de estas ideas.

[424: pto1] Selección y análisis de puntos de operación en sistemas complejos: ideas generales

Materiales: [PtOper.pdf]

****  18:57

[YouTube ►]

Una de las etapas importantes en control de sistemas complejos multivariables es la selección del punto “óptimo” de operación (discusión genérica sobre dichas etapas en vídeos [421], [422] y [423]).


Punto de operación óptimo: índices económicos + restricciones (límites de actuadores, seguridad, volumen de producción).

Sensibilidad del punto de operación óptimo ante perturbaciones no medibles: si el punto donde “deberían estar” las variables depende mucho de elementos desconocidos, de nada sirve fijar una referencia incorrecta que sea seguida con gran precisión.

Necesario analizar propiedades, por ejemplo sobre sistema linealizado y escalado, que sea “fácil de controlar” las salidas muy afectadas por perturbaciones, y que el coste de desviar los actuadores (coste del servomotor, energético, coste de materias primas, ...) compense el beneficio de reducir las desviaciones de salidas controladas... Nótese que, a este nivel, estamos hablando de **coste económico real**, y no de expresiones $x^2 + r \cdot u^2$ o similar en lo que se denomina “control óptimo” en los libros de texto de teoría de control. En caso de que el sistema sea difícil de controlar o el coste de reducir las desviaciones sea prohibitivo: cambiar punto de operación o rediseñar el proceso.

En resumen, en control de sistemas complejos se debe (a) seleccionar punto de operación, (b) seleccionar qué variables necesitan ser controladas y con qué actuadores y luego (c) seleccionar sensores para estimar el *estado interno* del sistema. Una vez estas decisiones han sido tomadas adecuadamente, sólo entonces es cuando el diseño de controladores tiene sentido: es la última etapa (d), que rara vez puede solventar problemas derivados de malas decisiones en etapas anteriores.

Un “experimento mental” sobre un problema (simplificado) de climatización discutiendo los conceptos genéricos aquí contemplados para ese caso particular se aborda en los vídeos [425] y [426].

[425: hvacop1] Selección óptima del punto de operación: ejemplo instalación de climatización (idealizado)
***  16:05

[\[YouTube ▶\]](#)

* [LINK TO ENGLISH VERSION](#)

Este vídeo presenta las decisiones a tomar de cara a optimizar un punto de operación en un sistema de aire acondicionado, como un experimento mental sobre algo concreto de las ideas genéricas del vídeo [424].


Realmente el planteamiento es idealizado, en el sentido de que, por un lado, nadie usaría software de optimización multicriterio para elegir la temperatura del mando del aire de su casa y, por otro lado, el confort depende de muchos factores además de temperatura con lo que el estudio exhaustivo del punto de operación óptimo en una oficina o nave con múltiples trabajadores requiere conocimientos de expertos en la materia que yo no tengo.

No obstante, este proceso de aire acondicionado permite entender intuitivamente, sin fórmulas adicionales, los conceptos de:

- Optimización multicriterio (compromiso entre requerimientos contradictorios),
- Solución dominada,
- Índice de coste, curva de igual preferencia,
- Significado de la convexidad de la función de coste en este problema concreto, que la hace una opción aconsejable,
- Punto de operación óptimo compromiso entre confort y consumo energético una vez introducida la física del fenómeno.

Este caso de estudio continúa en el vídeo [426], donde se consideran posibles perturbaciones y se discute la elección de determinadas variables (temperatura o potencia) como “variables controladas” a mantener constantes.

13.2 Selección de actuadores y variables controladas primarias

[426: hvacop2] Selección óptima de pto. op. y variables controladas: instalación de climatización
***  18:33

[\[YouTube ▶\]](#)

* [LINK TO ENGLISH VERSION](#)

Este vídeo es continuación del vídeo [425], donde se presentaban las decisiones a tomar de cara a optimizar un punto de operación en un sistema (idealizado) de aire acondicionado.

Aquí continuamos discutiendo las ideas genéricas del vídeo [424], esta vez pensando en qué debemos considerar a la hora de elegir cierta señal interviniente en el proceso como “variable controlada” (ponerle una referencia para mantenerla constante frente a perturbaciones que intentan cambiarla).

La idea básica es que una variable es candidata a ser elegida como “controlada” si su valor óptimo (de acuerdo a ciertos criterios y compromisos de diseño) se mantiene constante aunque determinados factores exógenos (perturbaciones) cambien.

Se presentan tres “experimentos mentales” ante posibles cambios de temperatura exterior y diferentes prioridades en el coste: en uno de ellos la temperatura es una buena opción como variable controlada, en otro es la potencia la que se recomendaría mantener constante, y en otro no es ninguna de las dos señales y se requeriría, por tanto, una etapa de “optimización” que generara el punto de funcionamiento (midiendo la temperatura exterior, una especie de prealimentación calculada mediante optimización, generalizando pues el concepto y el ámbito de aplicación de las ideas de prealimentación/feedforward).

NOTA: este vídeo es, realmente, una forma muy complicada de decir algo muy sencillo: "si el usuario prioriza confort, la variable controlada debe ser temperatura", "si prioriza ahorro, que vaya a potencia constante cuando el confort sea inaceptable". Pero la idea es entender el léxico usado en problemas de optimización de procesos más complejos, no creo que nadie haga nada de lo que aquí se cuenta en la climatización de su salón, al menos no conscientemente.

[427: saplant]

Selección de actuadores y variables controladas: planteamiento del problema

** 10:22

Materiales: [SelectLP.pdf]

[YouTube ▶]

Planteamiento del problema de selección de actuadores y variables que se desea controlar:

- Clasificación de las salidas del sistema: variables controladas primarias / secundarias / no controladas.
- Condicionantes: saturación en amplitud de actuadores, ancho de banda (rapidez) de actuadores, rapidez en cambios de referencia, amplitud y ancho de banda de perturbaciones sobre salidas a controlar.
- régimen permanente (constante) / transitorio.

[428: sarefp1]

Selección actuadores y variables controladas: enfoque poliédrico seguimiento referencias, plantas cuadradas

*** 08:37

Materiales: [saref.zip]

[YouTube ▶]

Este vídeo describe las condiciones que la matriz de transferencia G debe cumplir para poder conseguir, sin saturar actuadores, unos incrementos de salida iguales o superiores a unos prefijados.

Se analiza sólo el caso constante (con matriz de ganancia estática). En el caso de plantas cuadradas, se puede probar que si todas las combinaciones de los valores "extremos" (vértices ν_i) de referencias solicitadas pueden conseguirse sin saturar actuadores ($u_i = G^{-1}\nu_i$ no rebasa los límites en coordenadas incrementales), entonces todas las combinaciones intermedias tampoco saturarán.

Se presenta un ejemplo numérico en la segunda mitad del vídeo. En el vídeo se hace referencia a manipulación de poliedros. Dependiendo del background del lector, si se requiere más conceptos preliminares sobre poliedros de cara a comprender los desarrollos, se recomienda consultar el vídeo [913].

El caso de plantas no cuadradas con más actuadores que variables controladas se analiza en el vídeo [429], continuación de éste.

[429: sarefp2]

Selección actuadores y variables controladas: enfoque poliédrico seguimiento referencias, exceso de actuadores

**** 11:35

Materiales: [saref.zip]

[YouTube ▶]

Este vídeo describe las condiciones que la matriz de transferencia G debe cumplir para poder conseguir, sin saturar actuadores, unos incrementos de salida iguales o superiores a unos prefijados. Se analiza sólo el caso constante (con matriz de ganancia estática). El caso de plantas con igual número de referencias y actuadores (plantas cuadradas) se analizó en el vídeo [428], cuya visualización se recomienda antes de éste, así como el vídeo [913] sobre poliedros.

En el caso de plantas con exceso de actuadores (eso es bueno, en principio), se presenta una solución aproximada (pero muy útil en la práctica) mediante la pseudoinversa G^\dagger (pinv):


si todas las combinaciones extremas ν_i verifican que $G^\dagger \nu_i$ no satura, tampoco lo harán las intermedias. Como la pseudoinversa resuelve un problema de mínimos cuadrados, evidentemente sería necesario un escalado previo a unidades “homogéneas” para que $u_1^2 + \dots + u_p^2$ tuviera el adecuado significado físico/tecnológico.

En el caso de más actuadores que variables controladas bajo consideración, el cálculo exacto se hace comprobando si un poliedro $\{\nu_i = Gu, Ru \leq l\}$ es *no vacío*. Una forma podría ser mediante los comandos **linprog** o **quadprog**: resolviendo determinado problema de programación lineal o cuadrática, si no dan error, es que el poliedro no era vacío.

Se presentan ejemplos numéricos y código Matlab asociado en la parte final del vídeo.

[430: sap1] Selección variables controladas/manipuladas para cancelar perturbaciones: cancelación total (RP), enfoque poliédrico

Materiales: [SelectLP.pdf]

***  07:52


[YouTube ▶]

Este vídeo comienza con una revisión de planteamiento del problema (vídeo [427]) y fórmulas de prealimentación (vídeo [475]).

El video determina si se pueden cancelar en su totalidad perturbaciones en un determinado rango con un cierto rango de actuadores, con comprobación de vértices de efecto de perturbaciones, que multiplicados por la inversa de la planta no saturen. Incluye ejemplo Matlab.

[431: sap2] Selección variables controladas/manipuladas para cancelar perturbaciones: cancelación parcial (RP), enfoque poliédrico

Materiales: [SelectLP.pdf]


****  07:47

[YouTube ▶]

Este vídeo discute la cancelación parcial de perturbaciones: determinar si existe una acción de control que no sature que lleve el error a un cierto intervalo (no a cero). Resolución como factibilidad de un problema de programación lineal (determinar si ciertos poliedros son no vacíos). Incluye ejemplo Matlab.

[432: sap3] Selección variables controladas/manipuladas: conclusiones enfoque poliédrico (rég. perm.)

Materiales: [SelectLP.pdf]

****  06:41


[YouTube ▶]

El vídeo presenta la unificación en un único problema $e = Gu + [-I \ H] \begin{pmatrix} r \\ d \end{pmatrix}$ de rechazo de señales exógenas tanto del seguimiento de referencias como de cancelación de perturbaciones.

El vídeo termina con algunas consideraciones prácticas, y la relación de los enfoques propuestos con el control predictivo (que usa **quadprog** explícitamente).

[433: sa3ref] Selección de actuadores y variables a controlar: enfoque SVD (seguimiento de referencia, rég. permanente)

Materiales: [SelectLP.pdf]

***  10:26

[YouTube ▶]

Este video presenta el enfoque basado en descomposición en valores singulares (SVD) en régimen permanente de selección de actuadores y variables a controlar.

Discute las siguientes ideas:

- Resumen planteamiento del problema.

- Escalado a ± 1 de entradas, variables controladas y perturbaciones, siguiendo las ideas de mínimos cuadrados ponderados para escalado de salidas (vídeo [929]) y entradas (vídeo [930]).
- Criterio de ganancia **mínima** de G_{esc} mayor de 1 para seguimiento de referencias. Equivalencia $\underline{\sigma}(G_{esc}) = 1/\overline{\sigma}(G_{esc}^{-1})$.
- Diferencias con el enfoque poliédrico (geometría de *elipses* versus geometría de *poliedros*) de vídeos anteriores.

Ejemplos numérico de los conceptos aquí discutidos aparecen en los vídeos [434] y [446].

La cancelación de perturbaciones (continuación teórica) se aborda en el vídeo [435].

[434: sa3refex]

Selección de actuadores SVD (seguimiento de referencia, rég. permanente): ejemplo

*** 06:23

Materiales: [SelactLP.pdf]

[YouTube ▶]

Este vídeo desarrolla un ejemplo de selección de actuadores basada en SVD para un modelo de proceso de 4 variables controladas y 7 posibles variables manipuladas, mediante el SVD, según la teoría descrita en el vídeo [433], resumida en el primer minuto de éste.

Primero se analiza la controlabilidad entrada-salida de la matriz completa, y luego se analiza un subconjunto de 4 actuadores (selección de columnas) para comprobar si serían suficientes para controlar la planta en régimen permanente sin saturar.

Otro ejemplo de análisis (en frecuencia) de estas propiedades aparece en el vídeo [446].

[435: sa3pc]

Selección de actuadores SVD: cancelación total de perturbaciones, condicionamiento

*** 06:07

Materiales: [SelactLP.pdf]

[YouTube ▶]

Este video presenta el enfoque basado en descomposición en valores singulares (SVD) en régimen permanente de selección de actuadores y variables a controlar, continuando con las ideas presentadas en el vídeo [433].

Aquí se aborda el problema de dimensionar actuadores para rechazo (total) de perturbaciones, basado en la fórmula de la prealimentación lineal (detalle en vídeo [475]).

En concreto, se propone un criterio de ganancia **máxima** de $G^\dagger G_{pert}$ para cancelación total de perturbaciones († denota matriz pseudoinversa).

Posteriormente, se recuerda que debe comprobarse que G_{esc} debe tener un condicionamiento numérico adecuado (revisar vídeo [786]), tanto para seguimiento de referencia como para rechazo de perturbaciones.

La parte final del vídeo discute un ejemplo numérico (Matlab) con un modelo de perturbaciones.

La generalización del enfoque SVD a cancelación parcial de perturbaciones (errores bajo unos umbrales no nulos) se aborda, basada en un ejemplo Matlab concreto, en el vídeo [449].

[436: satrans]

Selección de variables controladas y manipuladas: caso transitorio (seguimiento referencias/cancel. total perturbaciones)

**** 11:20



Materiales: [SelactLP.pdf]

[YouTube ▶]

Este vídeo utiliza la respuesta en frecuencia (la matriz de respuesta en frecuencia $G(j\omega)$) para añadir consideraciones de rapidez / ancho de banda a los problemas de los videos anteriores [428]–[435].



Se revisan ideas sobre:

- Valores singulares de $G(j\omega)$: interpretación de ganancias máxima y mínima, combinaciones de senoides de la misma frecuencia pero distinta amplitud y desfase con el mayor/menor efecto.
- Aplicación a seguimiento de referencias y rechazo exacto de perturbaciones con un cierto ancho de banda.
- Ejemplo numérico (Matlab). Relación aproximada ancho de banda / tiempo de establecimiento (o de subida, si oscilatorio). De hecho, un ejemplo de esta relación se aborda en la parte final del vídeo [273].

[437: satransP] Selección actuadores y variables a controlar: cancelación parcial *****  07:15
perturbaciones, régimen transitorio
Materiales: [SelectLP.pdf] 

Este vídeo relaja los requerimientos de cancelación de perturbaciones en la frecuencia de cancelación *total* discutida en el vídeo [436] a cancelación *parcial*. Ahora, además de usar los valores singulares de $G(j\omega)$, deben usarse los vectores asociados a la descomposición SVD completa $G(j\omega) = U(j\omega)S(j\omega)V(j\omega)^H$.

Nota: Este enfoque, tomado del libro de Skogestad *Multivariable Feedback Control*, es conservativo y, realmente, poco aconsejado... con requisitos de cierto tamaño de error en frecuencia y tamaño de actuadores, la forma recomendada de hacerlo con la geometría de las esferas/elipses (la única válida en frecuencia) es plantear un problema \mathcal{H}_∞ , como se discute en profundidad en otros materiales de la colección (p.ej., metodología *mixed sensitivity*). De hecho, todas las discusiones SVD en estado estacionario o/y transitorio sobre en los vídeos previos a este no son más que una aproximación al problema formalmente más general del control \mathcal{H}_∞ sobre plantas generalizadas... Didácticamente, es muy conveniente comenzar exponiendo todo lo de los vídeos anteriores a este sobre selección de variables controladas y manipuladas, referencias, perturbaciones, etc. porque ‘SVD’ o ‘quadprog’ son más fáciles de entender y ayudan a comprender mejor esos contenidos posteriores... Formalmente, todo el enfoque SVD es posiblemente prescindible si se empezase directamente con \mathcal{H}_∞ .

[438: sa4conc] Selección de actuadores y variables a controlar: conclusiones ***  06:23
finales
Materiales: [SelectLP.pdf] 

Este vídeo presenta las conclusiones globales de todo el desarrollo en los videos sobre el tema de dimensionamiento de actuadores y selección de variables controladas, tanto desde un punto de vista matemático/formal como desde un punto de vista de ingeniería de control para aplicaciones (básicamente, hay que añadir consideraciones económicas en las decisiones). También se recomienda visualizar las conclusiones genéricas al final del caso de estudio del vídeo [449] (minuto [11:10]).

Importante: Los resultados en esta sección han considerado un análisis “preliminar”, “rápido”, basado en el modelo escalado del proceso. La comprobación sobre si se pueden seguir o no ciertas referencias “transitorias” realmente es una aproximación: se está usando el régimen *estacionario* senoidal (comando **sigma**), y las referencias y perturbaciones no tienen por qué tener un escalado “constante” a todas las frecuencias: la amplitud esperada de las referencias o perturbaciones puede

estar relacionada con su “rapidez” (por ejemplo, esperar referencias de amplitud 1 a frecuencias menores a 2 rad/s, pero de amplitud 0.1 a frecuencias de 2 rad/s); una generalización del escalado dará lugar a lo que se llama “ponderación en frecuencia” (en el ejemplo, amplitud de referencia por debajo de la amplitud de $W_{ref} = 0.1 \cdot (s + 20)/(s + 2)$), tanto de referencias como perturbaciones (que se agruparan como “entradas externas”). Los actuadores también podrían tener una capacidad de movimiento más limitada a frecuencias altas y los objetivos de control también podrían ser diferentes a baja frecuencia (errores deseados “pequeños”) y a alta frecuencia (errores deseados “no tan pequeños”).

Por todo esto, desde un punto de vista formal, la validación completa sobre si se pueden seguir (o no) unos objetivos de control “sofisticados” se haría con el análisis basado en “planta generalizada ponderada” (vídeo [pgpo(19:09)]), cuyos pesos están estrechamente relacionados con las ideas de escalado aquí propuestas, según se detalla en la Sección 19.3.2.

Control con saturación durante una fracción importante del transitorio: Considerando saturación, los procesos pueden “acelerarse” más de lo que se calcula basado en el cruce con 0dB de la ganancia escalada. El control para procesos lineales que puede saturar es el control **predictivo**. Consideraciones sobre dicha cuestión se abordan en el vídeo [dacta(09:31)].

Procesos inestables y de fase no mínima. Es de especial interés mencionar el caso de procesos inestables y de fase no mínima donde los anchos de banda indicados por el cruce de $\underline{\sigma}(G(j\omega))$ con los 0 dB peca en muchos casos de excesivamente optimista, por dos razones:

- En procesos *inestables* se requiere de una cierta actividad del regulador “mínima” para estabilizar; por ello, el margen hasta saturación que queda “libre”, para poder estabilizar y *además* cancelar perturbaciones o reducir el error ante referencias, se reduce respecto al de un proceso con el mismo diagrama de valores singulares pero que fuera estable. La mínima actividad de la acción de control para estabilizar es analizada en el vídeo [minust(10:54)], mediante el enfoque de planta generalizada \mathcal{H}_∞ arriba mencionado.
- En procesos de *fase no mínima* (o retardo), evitar cancelaciones polo-cero inestable obliga a **no** poder **cancelar** los ceros de fase no mínima; intentar un tiempo de establecimiento menor al tiempo donde la respuesta del sistema tiene signo contrario al de la ganancia resulta complicado (cancelar el cero requeriría una amplitud de control que tendería a infinito, lo cual es imposible con actuadores limitados). Si el proceso es, simultáneamente, inestable y de fase no mínima, entonces los problemas se agravan. Esto será discutido en secciones posteriores, en concreto la Sección 19.4.

13.2.1 Ejemplos adicionales

Sistema térmico orden 4

[439: termsa1]

Sistema térmico orden 4: dimensionamiento de actuadores (rég. perm., poliedros)

Materiales:

10:25

[CÓD.: termexampleSelect.mlx]
[PDF]
[YouTube ►]

Este vídeo plantea el problema de selección (dimensionamiento) de actuadores en el sistema térmico de orden 4 cuyo modelo se discute en el vídeo [18], y cuya simulación en Matlab y análisis de propiedades se aborda en el vídeo [274].

El objetivo es conseguir demostrar que, sin saturar, pueden efectuarse maniobras que incrementen las temperaturas de las habitaciones o/y que cancelen unos ciertos incrementos de temperatura ambiente exterior.

Este vídeo plantea el estudio en régimen permanente (matriz de ganancia estática), enumerando los vértices (valores extremos) de las referencias y perturbaciones y estudiando si las acciones de control necesarias (a partir de $u = G^{-1}(r - G_{pert}T_a)$) saturan o no.

El estudio basado en descomposición en valores singulares (que puede, aproximadamente, usarse en transitorios –cosa que el enfoque basado en poliedros no permite–) se aborda en el vídeo [440].

[440: termsa2]

Sistema térmico orden 4: dimensionamiento de actuadores
(SVD, transitorio)

Materiales: [[CÓD.:](#) termexampleSelect.mlx] [[PDF](#)]

**** 14:17

[[YouTube](#) ►]

Este vídeo plantea el problema de selección (dimensionamiento) de actuadores en el sistema térmico de orden 4 cuyo modelo se discute en el vídeo [18], y cuya simulación en Matlab y análisis de propiedades se aborda en el vídeo [274].

El objetivo es conseguir demostrar que, sin saturar, pueden efectuarse maniobras que incrementen las temperaturas de las habitaciones o/y que cancelen unos ciertos incrementos de temperatura ambiente exterior.

El vídeo [439] plantea el estudio en régimen permanente (matriz de ganancia estática), enumerando los vértices (valores extremos) de las referencias y perturbaciones y estudiando si las acciones de control necesarias (a partir de $u = G^{-1}(r - G_{pert}T_a)$) saturan o no (geometría de *poliedros*).

En este vídeo se plantea el mismo problema, esta vez resuelto mediante la descomposición SVD de G , G^{-1} , $G^{-1}G_{pert}$, etc.

Obviamente, el SVD implica que la geometría que estamos considerando es la de los *elipsoides* (una vez escalados, las esferas de radio 1). La parte final del vídeo considera las diferencias entre unas geometrías y otras, y el factor \sqrt{n} siendo n la dimensión del vector de entradas (referencias, perturbaciones) necesario para contemplar el caso de que todos los elementos alcancen simultáneamente sus valores extremos.

Sistema térmico orden 11 (horno lineal)

[441: forn1]

Caso de estudio: Modelado y analisis controlabilidad
entrada/salida de sistema horno lineal orden 11

Materiales: [[CÓD.:](#) modelohornoc.mlx] [[PDF](#)]

*** 15:49

[[YouTube](#) ►]

Este vídeo modela un horno de calentamiento de una barra de cierto material con tres actuadores: radiador en parte izquierda, radiador en parte derecha y actuador 3 que distribuye de forma uniforme potencia calorífica.

*El modelo es puramente académico, inventado, y no pretende ningún parecido con un horno real.

Si estás empezando en estos temas, quizás te interese visionar el vídeo [584] donde se revisan los mínimos cuadrados y distintas opciones de tener p bajo control' un horno muy similar al que aquí se estudia.

Tras obtener un modelo en representación interna, se analiza su respuesta temporal y su ganancia, para comprobar que se corresponde con el comportamiento esperado intuitivamente.

A partir del minuto [06:30], se calcula la descomposición SVD de la matriz de ganancia estática. Se identifican los espacios columna y nulo de la misma, y se analiza el significado de las maniobras (componentes principales) ortogonales en las que se descompone el conjunto de temperaturas alcanzable con los actuadores. Por último, se evalúan los valores singulares en función de la frecuencia (sigma) y se sacan conclusiones sobre el ancho de banda o tiempo de subida/establecimiento conseguible con controladores a ser diseñados en el futuro.

El modelo de horno aquí discutido es utilizado para diseñar controladores por diversas técnicas en vídeos posteriores: control PI mediante desacoplamiento SVD en el vídeo [498], control \mathcal{H}_2 en el vídeo [666] y control \mathcal{H}_∞ en el vídeo [667].

Sistema térmico experimental

[442: icscpl11] ICSC-PL1 (1): descripción prototipo horno

***  11:12

[YouTube ▶]

Este es un vídeo para mis estudiantes de la asignatura "Identificación y Control de Sistemas Complejos" en la ETSII-UPV, para ayudar a la realización de la PL1 y el trabajo académico 1 asociado.

Describe el prototipo de horno (funcionamiento y detalles de construcción) sobre el que se tomaron datos experimentales y se hicieron experimentos de identificación y control, discutidos en otros vídeos (en concreto, los [443] y [444]).

[443: icscpl12] ICSC-PL1 (2): Descripción de experimentos y formato de datos

***  14:45

[YouTube ▶]

Este es un vídeo para mis estudiantes de la asignatura "Identificación y Control de Sistemas Complejos" en la ETSII-UPV, para ayudar a la realización de la PL1 y el trabajo académico 1 asociado.

Este vídeo es continuación del [442] en el que se describe la planta física.

Aquí se describe el experimento de respuesta escalón sobre el que se determinó punto de funcionamiento y respuesta "incremental" alrededor de él, así como el formato de los datos del fichero Experimento.mat proporcionado para la práctica y el trabajo.

Un análisis preliminar de algunos resultados de identificación y controlabilidad entrada-salida se aborda en el vídeo [444], continuación de éste.

[444: icscpl13] ICSC-PL1 (3): Discusión preliminar sobre resultados de identificación y controlabilidad E/S

***  14:58

[YouTube ▶]

Este es un vídeo para mis estudiantes de la asignatura "Identificación y Control de Sistemas Complejos" en la ETSII-UPV, para ayudar a la realización de la PL1 y el trabajo académico 1 asociado.

Este vídeo es continuación de los vídeos [442] y [443], en los que se describe la planta física y los experimentos y datos obtenidos.

Aquí se describe el resultado de la identificación de algunas de las funciones de transferencia individuales que conforman el modelo 6x6 global, se discute su dimensión, orden, y la conveniencia de su reducción (al menos para diseños centralizados de realimentación del estado y observadores). También se detalla la interpretación del SVD de la ganancia estática (escalada) y del diagrama de valores singulares para estimar la rapidez con la que las temperaturas pueden ser modificadas en el proceso.

Otros ejemplos

[445: dacta] Análisis de controlabilidad entrada-salida proceso con 2 actuadores de diferente ancho de banda (Matlab)

***  09:31

Materiales: [Cód.: cascadaSobraActPID.mlx]

[YouTube ▶]

En este vídeo se analiza la controlabilidad entrada-salida de un proceso con un actuador rápido pero de poca ganancia y otro segundo actuador de cuatro veces más ganancia pero ocho veces más lento. Las referencias de pequeña amplitud podrían ser seguidas con rapidez, incluso usando el actuador rápido únicamente; las de amplitud más elevada requieren ser más lentas y el uso del segundo actuador.

La parte final del vídeo aborda de forma cualitativa la posible mejora en prestaciones si se fuese a usar un controlador donde los actuadores saturaran durante una parte significativa del transitorio; esto puede ser por haber intencionalmente subido las ganancias de los PIDs (que será lo más frecuente en la práctica) o por usar control predictivo o no lineal que contemplan estas saturaciones explícitamente en sus metodologías. En ese caso, el límite de rapidez vendría dado por el cruce de la respuesta temporal ante escalón con el valor de referencia (en orden superior habrá que “frenar” adecuadamente, claro). Hay que aclarar que en el caso multivariable con acoplamiento, o en procesos inestables, la conmutación entre saturación/linealidad de varios actuadores en diferentes instantes y diferentes direcciones resulta mucho más complicada de analizar. El vídeo [446] en su parte final también analiza la respuesta ante escalón en los componentes principales SVD de una planta multivariable, complementando el punto de vista aquí presentado.

Nota: Este tipo de configuración de actuadores es manejada en control descentralizado (PID) mediante la estructura en cascada con actuador extra, descrita en el vídeo [508] y, específicamente, la simulación en este mismo proceso de esta estructura de control se aborda en el ejemplo Matlab del vídeo [509].

[446: props3x3]
Análisis de propiedades y controlabilidad entrada-salida de un sistema 3x3

 13:48

Materiales:
[[Cód.:](#) AnalisisPropiedadesSVD3.mlx] [[PDF](#)]
[[YouTube](#) ►]

Este vídeo discute el análisis de propiedades de un sistema de 3 entradas y 3 salidas dado en matriz de transferencia G . Suponiendo que los estados de los elementos de dicha matriz de transferencia no tienen significado físico, se obtiene una realización mínima. Se supone que la planta G está ya escalada con entradas y salidas en el intervalo de extremos ± 1 .

Primero se analizan los “Hankel singular values” (gramiano de realización equilibrada), para determinar si el sistema podría reducirse (la conclusión es que sí, a orden 4, con muy poco error), según la teoría del vídeo [295].

Después se analiza la controlabilidad entrada-salida, mediante diagrama de valores singulares en frecuencia y descomposición SVD de la ganancia estática $G(0) = USV^T$, según lo discutido en el vídeo [433]. Se identifican dos maniobras controlables y una que no, así como los anchos de banda máximos a los que se podrán ejecutar dichas dos maniobras.

Con las matrices del SVD de la ganancia, se propone un cambio de variable en entradas y salidas de modo que tengamos $G^* = U^T \cdot G \cdot V$. Ello produce un sistema cuya ganancia estática es la matriz de valores singulares S , diagonal. La idea origina la metodología denominada *desacoplamiento SVD*, ver vídeo [489]. La respuesta ante escalón de dicho sistema G^* permite también estimar la rapidez con las que podrá alcanzarse una amplitud unidad (objetivo) en cada una de las maniobras (componentes principales) analizadas.

El vídeo continua planteando las distintas opciones para diseñar controladores dado que no se pueden controlar las tres salidas iniciales con la amplitud deseada. La parte final analiza el efecto de un cero de fase no mínima que tiene el proceso, que se observa en la segunda maniobra del SVD, y que limitará las prestaciones alcanzables.

En resumen, el análisis de rapidez/dimensionamiento conseguible mediante el **sigma-plot** es meramente “orientativo”: es **aproximado** (reg. est. senoidal, geometría de elipses), **conservador** si las técnicas de control predictivo son permitidas (podría conseguirse más rapidez de la que calculamos), pero **demasiado optimista** (los bucles serán mucho más lentos del resultado de este cálculo) en caso de procesos inestables, o de fase no mínima, o con retardo. También los errores de modelado disminuirán los objetivos de prestaciones “recomendables” (los controladores demasiado rápidos son poco tolerantes a errores de modelado, compromiso especificaciones robustez, Sección 25.2).

Caso de estudio sistema 2x3

[447: sa23a] Controlabilidad entrada/salida (estático) 2x3, caso estudio: [1]
planteamiento y enfoque SVD error cero
Materiales: [CÓD.: [selact2x3ce.mlx](#)] [PDF]

*** 09:36

[YouTube ►]

Este vídeo (primero de tres) presenta un caso de estudio estático en un problema de cancelación de perturbaciones $y = Gu + G_p p$ (realmente, como $G_p = -I$, se trata de un seguimiento de referencias), donde G tiene dos salidas (vbles. controladas) y tres entradas (vbles. manipuladas). Se darán unos ciertos umbrales de amplitud de perturbaciones (\equiv referencias en este caso particular), de saturación de actuadores, y umbrales de error deseado. El vídeo detalla la solución (SVD) para el caso donde las perturbaciones estén en cierto elipsoide, y se desee error cero: se hará el adecuado escalado y pseudoinversa; la teoría subyacente se presenta en los vídeos [435] y [433]. El resultado sale *factible*, porque la ganancia máxima de perturbaciones a control (escalados) con la fórmula de prealimentación sale menor a la unidad.

Comparativa con otros escenarios de error no necesariamente cero (cancelación parcial de perturbaciones) y técnicas basadas en poliedros se llevan a cabo en los vídeos [448] y [449], continuación de éste.

[448: sa23b] Controlabilidad entrada/salida (estático) 2x3, caso estudio: [2]
enfoque poliedros (MPT3, quadprog)
Materiales: [CÓD.: [selact2x3ce.mlx](#)] [PDF]

**** 14:49

[YouTube ►]

Este vídeo (segundo de tres) presenta un caso de estudio estático en un problema de cancelación de perturbaciones $y = Gu + G_p p$, ante ciertos umbrales de amplitud de perturbaciones (\equiv referencias), de saturación de actuadores, y umbrales de error deseado. La solución para error cero (cancelación total de perturbaciones) basada en valores singulares se aborda en el vídeo [447].

En el segundo caso a considerar aquí, se plantea un enfoque basado en poliedros y vértices del conjunto de perturbaciones esperables; la teoría subyacente se presenta en los vídeos [428] y [429].

En primer lugar, se calcula la acción de control para cancelación total de mínima norma mediante $\text{pinv}(G) * \xi$, siendo ξ cada uno de los 4 vértices del poliedro de perturbaciones dado como objetivo para cancelar (bueno, se hace escalando adecuadamente a ± 1). El resultado no es factible y, por tanto, debe comprobarse la factibilidad (que exista acción de control que no sature) planteando, por ejemplo, un problema de programación cuadrática (quadprog). Antes de plantear ese problema, se representa gráficamente el poliedro de referencias factibles que pueden ser canceladas (se grafican dos poliedros, el de error cero y el de error por debajo de cierto umbral –obviamente, el segundo de los poliedros es más grande–); para ello se usa código de la toolbox MPT3, pero realmente si no se desea “ver” ese conjunto, con quadprog es suficiente, como se comentó en el vídeo [429]. El código presentado comprueba que el problema planteado *sí* es factible.

Un escenarios de error no necesariamente cero (cancelación parcial de perturbaciones) basado en mínimos cuadrados ponderados y SVD se plantea en el vídeo [449], continuación de éste, así como un análisis comparativo entre los enfoques poliédrico y SVD.

[449: sa23c] Controlabilidad entrada/salida (estático) 2x3, caso estudio: [3]
enfoque SVD error no nulo, comparación y conclusiones
Materiales: [CÓD.: [selact2x3ce.mlx](#)] [PDF]

**** 14:49

[YouTube ►]

Este vídeo (tercero y último) presenta un caso de estudio estático en un problema de cancelación de perturbaciones $y = Gu + G_p p$, ante ciertos umbrales de amplitud de perturbaciones (\equiv referencias), de saturación de actuadores, y umbrales de error deseado. La solución para error cero (cancelación total de perturbaciones) basada en valores singulares se aborda en el vídeo [447], y el enfoque poliédrico (tanto para cancelación total como parcial) se aborda en el vídeo [448].

En este último caso, se plantea un enfoque basado en elipsoides (pseudoinversas, svd) asociado a minimizar $J = \|error_{esc}\|^2 + \|u_{esc}\|^2$; si $J < 1$ cuando las perturbaciones (escaladas) están en una circunferencia de radio 1, entonces todos los errores y acciones de control escaladas serán menores a 1. Con manipulaciones sencillas, minimizar J es un problema de mínimos cuadrados clásico, que se resuelve con una pseudoinversa y se comprueba la factibilidad del problema con un SVD de cierta matriz de ganancia de “error”.

Las direcciones asociadas a un valores singulares de J pequeños son maniobras “fáciles”, donde podría permitirse una magnitud de perturbaciones más grande. La parte final del vídeo dibuja gráficamente ese elipsoide “más grande” de perturbaciones donde también se garantiza factibilidad y lo compara con el conjunto poliédrico factible obtenido con el código MPT3 discutido en el vídeo [448].

La parte final del vídeo (minuto [11:10]) presenta conclusiones generales sobre ventajas e inconvenientes de enfoques “poliédricos” versus “pinv+SVD” que este caso de estudio ha ayudado a comprender mejor (se aconseja también ver las conclusiones del vídeo [438]).

En particular, es importante resaltar que minimizar $\|error_{esc}(j\omega)\|^2 + \|u_{esc}(j\omega)\|^2$ da lugar a lo que se conoce como problema “mixed sensitivity” del control \mathcal{H}_∞ , ver vídeo [812].

Caso de estudio sistema 2x3, otro proceso

[450: sacerfl] Selección vbles. manipuladas y controladas: seguimiento referencias, SVD, ejemplo Matlab (1)

Materiales: [CÓD.: solParc123P1part1.mlx] [PDF]

*** 12:58

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo, el primero de un caso de estudio con un total de 5 vídeos, plantea el problema de determinar si un proceso $y = Gu$ tiene bien elegidas las variables controladas y y las manipuladas u para poder incrementar y en las cantidades requeridas para la aplicación (seguimiento de referencias). Se discute el caso de régimen permanente (matriz de ganancia estática).

En este primer vídeo, se plantea el problema, y se aborda su resolución mediante metodología SVD (geometría de las elipses, o de las esferas de radio 1 una vez escalado). En efecto, se revisa el escalado, la matriz de transferencia escalada, y se calcula la ganancia mínima (menor valor singular) comprobando que es superior a 1 y que el condicionamiento de 2.9 también es satisfactorio, por lo que se da el problema como factible.

Este razonamiento breve es lo que se requiere en los exámenes de lápiz y papel de mis asignaturas. Discusión adicional y representación gráfica de las elipses que subyacen a esta metodología se discute en el vídeo [451], continuación de éste.

Un caso de estudio SVD similar a este se discute en el vídeo [447], cuya visualización se aconseja.

[451: sacerfl] Selección vbles. manipuladas y controladas: seguimiento referencias, SVD, ejemplo Matlab (2)

Materiales: [CÓD.: solParc123P1part1.mlx] [PDF]

**** 15:45

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)


Este vídeo continua el caso de estudio que comenzó en el vídeo [450] que discutía el planteamiento del problema de selección/dimensionamiento de variables manipuladas y controladas para seguimiento de referencias.

En dicho vídeo se escalaba y comprobaba ganancia mínima y condicionamiento. En este vídeo se van a discutir conceptos adicionales como:

- Análisis U,S,V completo de la descomposición SVD. Se observan las maniobras fácil y difícil en direcciones de entrada y salida; también se observa el espacio nulo de la matriz de ganancia (dirección de u sin efecto en régimen permanente).
- Como el espacio nulo está muy alineado con el actuador 2, se repite el análisis suponiendo que dicho actuador no se incrementa; se comprueba que las ganancias y condicionamiento apenas cambia, con lo que el problema sigue siendo factible.
- Se representan gráficamente, en el dominio de la salida (escalada), el elipsoide alcanzable con la esfera unidad en actuadores, así como el círculo de radio 1 y el de ganancia mínima. Se comprueba que, como la ganancia mínima es $\geq \sqrt{2}$ entonces la esfera unidad de entrada barre todo el cuadrado de vértices ± 1 en el dominio de la salida, garantizando la factibilidad del problema con geometría de poliedros (que se verá en el vídeo [452], continuación de este caso de estudio).

[452: sacerf3] [Selección vbles. manipuladas y controladas: seguimiento referencias, poliedros, ejemplo Matlab \(3\)](#)

Materiales: [[CÓD.:](#) solParc123P1part1.mlx] [[PDF](#)]

****  16:38

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

En este vídeo se aborda el mismo problema que en los vídeos anteriores [450] y [451] de este caso de estudio, que se revisa brevemente.

Luego, se aborda el problema como determinar si existe un u dentro de límites que verifique $Gu = v_i$, siendo v_i cada uno de los cuatro vértices extremos del rectángulo de salidas incrementales deseadas. Se hace con [quadprog](#), justificando que como sólo nos interesa factibilidad, el índice de coste puede ser cualquiera, por ejemplo, minimizar $\|u\|^2$.

A partir del minuto [07:45] se discute un par de observaciones sobre la geometría de poliedros:


Primero, que para aproximarse al problema de pseudoinversas (escalado), la matriz H del comando quadprog debería ser $\text{inv}(E_u)^2$; así, si la solución pseudoinversa/SVD es factible, coincidiría con el resultado de [quadprog](#).

Segundo, que haciendo “zoom” de los vértices de incremento de salida deseados hasta que el problema deje de ser factible, obtendremos un margen de exceso de potencia con una interpretación análoga a la ganancia mínima por encima de 1 del SVD.

La parte final del vídeo dibuja, en el espacio de salidas, los poliedros alcanzables con los incrementos de entrada disponibles y el deseado. Como el alcanzable es mayor que el deseado, el problema es factible. Los últimos minutos del vídeo superponen los resultados del enfoque SVD y del enfoque poliédrico del mismo problema, para poder hacerse una idea intuitiva de sus similitudes y diferencias.

[453: sacerf4] [Selección vbles. manipuladas y controladas: cancelación total perturbaciones, ejemplo Matlab \(4\)](#)

Materiales: [[CÓD.:](#) solParc123P1part1.mlx] [[PDF](#)]

****  15:59

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute el mismo caso de estudio que comenzó en el vídeo [450]. Aquí se considera el efecto de perturbaciones en $y = Gu + Hd$, y se busca determinar si los límites

de actuadores son suficientes para compensar totalmente el efecto de d sobre las salidas controladas y .

Para ello, primero se analizan los valores singulares de $-pinv(G) * H$, adecuadamente escalados.

En la segunda mitad del vídeo se plantea el problema en términos de geometría de poliedros, usando [quadprog](#) para comprobar su factibilidad.

La parte final del vídeo representa gráficamente el significado de las diferentes opciones consideradas para abordar el problema.

[454: sacerff5]	Selección vbles. manipuladas y controladas: cancel. parcial perturbaciones, poliedros (5)	****	CE	13:44
Materiales: [CÓD.: solParc123P1part1y2.mlx] [PDF]				[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

En este quinto vídeo del caso de estudio que comenzó en los vídeos [450], [451], [452] y [453] se aborda la cancelación parcial de perturbaciones en régimen permanente, con la geometría de poliedros.

Primero se generan los vértices de “error residual admisible” a partir de las especificaciones del problema y, con ellos, para cada uno de ellos se aborda la factibilidad de un problema de programación cuadrática que probaremos ejecutando [quadprog](#). Se discute brevemente el índice de coste más adecuado a minimizar, aunque esa discusión sólo sería relevante si el controlador en tiempo real sobre el proceso fuese a usar [quadprog](#) (control predictivo, usualmente) y las perturbaciones fueran medibles u observables. Como sólo buscamos factibilidad, realmente $H = I$, $f = 0$ nos valdría (y eso es lo único que se pediría en un examen de ‘lápiz y papel’ a mis estudiantes).

A partir del minuto 09:00 se discuten consideraciones adicionales opcionales; en particular, el margen de aumento de las perturbaciones de modo que el problema siga siendo factible, y la representación gráfica de los poliedros alcanzables en el espacio de las salidas y para una mejor comprensión de los conceptos (no se analiza el código en detalle, que se deja al lector).

[455: sacerff6]	Selección vbles. manipuladas y controladas: cancel. parcial perturbaciones, SVD (6)	****	CE	16:35
Materiales: [CÓD.: solParc123P1part1y2.mlx] [PDF]				[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este es el sexto vídeo del caso de estudio que comenzó en el vídeo [450]. Aquí se discute cancelación parcial de perturbaciones en régimen permanente con técnicas SVD; la geometría de los poliedros ([quadprog](#)) fue discutida en el vídeo [454].

Otro problema muy similar a éste se discute en el vídeo [449].

Básicamente, manteniendo los escalados de variables manipuladas y perturbaciones de problemas anteriores, ahora se discute un nuevo escalado de salida basado en el umbral máximo de error tolerable. Con ese nuevo escalado, se plantea el problema de calcular un u óptimo tal que $\|e_{esc}\|^2 + \|u_{esc}\|^2 \leq 1$ cuando $\|d\| \leq 1$. De ese modo, cada uno de los dos sumandos será obviamente menor a 1.

Como se trata de un problema de mínimos cuadrados, se resuelve con pseudoinversa y se comprueba el svd de la matriz resultante.

El caso de estudio concluye en el vídeo [456], que analiza el caso transitorio.

[456: sacerf7] **Selección vbles. manipuladas y controladas: régimen transitorio, sigma (7)**

Materiales: [[CÓD.: solParc123P1part1y2.mlx](#)] [[PDF](#)]

*** 10:56

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo concluye el caso de estudio que comenzó con el vídeo [450]. Aquí se discuten, en régimen transitorio (respuesta en frecuencia, **sigma**), el problema de seguimiento de referencias y el problema de cancelación total de perturbaciones.

La parte final del vídeo discute la posibilidad de incrementos de referencia y perturbaciones simultáneos, comparando con los resultados iniciales de sólo uno de los dos problemas.

El rechazo parcial de perturbaciones con límites de error dependientes de la frecuencia debería manejarse, posiblemente, con un control óptimo \mathcal{H}_∞ ; las gráficas de valores singulares aquí discutidas son solamente una primera aproximación fácil a tal problema.

13.3 Aplicación a robótica: elipsoides de manipulabilidad y fuerza

Elipsoide de manipulabilidad (velocidades)

[457: elipm1EN] **Manipulability of robotic arm (1): modelling, jacobian, singular value decomposition**

Materiales: [[CÓD.: ElipsRobot1ENG.mlx](#)] [[PDF](#)]

*** 16:33

[YouTube ▶]

This video introduces the concept of manipulability analysis, the discussion of which will continue in the video [458] (manipulability ellipsoid) and [459] (more examples and animations).

Specifically, the basic idea of the analysis is based on the robot Jacobian (the matrix that relates joint velocities to end-effector velocities) and its singular value decomposition $J = USV^T$, which gives rise to the “principal manoeuvres”. A theoretical analysis discussing the relationship between singular value decomposition (SVD) and ellipsoid geometry is further discussed in video [925].

This video makes a brief theoretical review of the Jacobian concept, models a robot with three articulated elements (planar one, no gravity), obtains said Jacobian, and analyzes its SVD. The case study continues in the videos referred to at the beginning of the description.

[458: elipm2EN] **Manipulability ellipsoid of robot arm (2): theory, example, polyhedron approach**

Materiales: [[CÓD.: ElipsRobot1ENG.mlx](#)] [[PDF](#)]

**** 17:52

[YouTube ▶]

This video continues the manipulability discussion started in video [457], which is summarized here in the first two and a half minutes.

In this video emphasis is placed on defining what is called the *manipulability ellipsoid*: the locus of end-effector velocities when joint velocities are on the unit sphere (after some scaling/normalizing).

After a theoretical review, said ellipsoid is represented graphically and the relationship of the principal maneuvers of the SVD of the scaled Jacobian with said ellipsoid is illustrated.


The final part of the video superimposes the manipulability ellipsoid with the locus of end-effector velocities when joint velocities are in the *cube* of vertices ± 1 . The ellipsoid approximates it reasonably, and allows the SVD principal manoeuvre interpretation

that the polyhedron approach does not easily allow. By contrast, the polyhedron is an exact representation of practical achievable speeds. Everything has its advantages and disadvantages.

Examples in other poses and animations will be illustrated in the video [459], a continuation of this one.

[459: elipm3EN] Manipulability ellipsoid of robot (3): further examples and animations

Materiales: [[CÓD.: animrobot.m](#)] [[PDF](#)]

***  12:23


[YouTube ▶]

This video graphically illustrates the principal maneuvers and manipulability ellipsoids discussed in the video [458] in different poses. Starting at minute [06:40], we also represent the principal maneuvers with an animation, to better understand their meaning.

Elipsoide de fuerzas

[460: elipf1EN] Jacobian: matrix gear ratio, effect on force/torque multiplication

Materiales: [ElipsRobot2Force.mlx]

****  20:22


[YouTube ▶]

This video discusses the role of the Jacobian matrix J in force/torque transmission in a robot arm (or in a generic mechanism, of course). In previous videos, chain rule justified that $\dot{r} = J\dot{q}$, being \dot{r} end effector speed and \dot{q} the joint speeds. Here, we discuss that $\tau = J^T f$ is the joint torque needed to compensate end-effector force f in order to keep the robot arm motionless.

We do it in two ways: a first one, sort of informal, via an energy (power) conservation argumentation, extracting intuition from a pinion-gear thought experiment; the second one is a more accurate account of what happens, deriving the same expression $\tau = J^T f$ through the Euler-Lagrange equations of motion.

[461: elipf2EN] Force ellipsoid in a robot (1): definition and basic examples

Materiales: [ElipsRobot2Force.mlx]

***  11:53

[YouTube ▶]


This video defines the *force ellipsoid* of a robotic manipulator. Said ellipsoid is the ellipsoid of end-effector forces that could be compensated to keep the manipulator immobile if the joint torques lie on the unit sphere.

The basic theoretical formulas are justified in the video [460], which you should watch before this one. Indeed, the fundamental idea is that the torque vector τ that compensates a force f in the end effector is $\tau = J^T f$, so $\tau^T \tau = f^T \cdot (J \cdot J^T) \cdot f \leq 1$ defines an ellipsoid (said “force ellipsoid”). The video introduces the possible scaling and draws the ellipsoid in different poses.

Theoretical SVD analysis continues in video [462], and the relationship between manipulability ellipsoids (velocities) and forces is discussed in video [463].

[462: elipf3EN] Force ellipsoid in a robot (2): principal maneuvers, Jacobian singular value decomposition

Materiales: [ElipsRobot2Force.mlx]


****  14:41

[YouTube ▶]

This video discusses the relationship of the *force ellipsoid* of a robotic manipulator (defined in the video [??]), with the singular value decomposition of the Jacobian (with an appropriate scaling).

The main maneuvers are obtained in the space of end-effector forces and normalized joint pairs.

The development is parallel, practically identical, to that of the principal maneuvers in the speed space discussed in the video [458]. In fact, it is so similar that (with unit scalings) the principal maneuvers are coincident (with inverse gain): the details of this important property are discussed in the video [463], which delves into the relationship between manipulability ellipsoids (velocities) and forces.

[463: elipf4EN]	Duality of force/manipulability ellipsoids in a robot (inverse behaviour)	***  14:43
Materiales: [ElipsRobot2Force.mlx]		[YouTube ▶]


This video concludes the analysis of the force and manipulability ellipsoids (object of videos [457], [461] and successive ones) of a robot. In this last video, we explore their duality: the main speed and force maneuvers have “inverse” amplitudes: if a maneuver multiplies angular velocity by 7 to obtain end-effector linear velocity, then the same maneuver divides joint torque by 7 to give end-effector force.

For this to be correct, operations must be performed on NON-scaled compatible physical units (m/s, rad/s, N, Nm).

This generalizes to “matrix version” the basic idea of many mechanisms (lever, gear): multiplying speeds by 7 in a lever divides forces by 7, for instance.


The orthogonality of the main SVD maneuvers means that the Jacobian of a robot can be interpreted as an XY table, with two different gear ratios given by the singular values.

13.4 Selección de la estructura de control

[464: ect1a]	Etapas iniciales de un problema de control de sistemas complejos, resumen.	***  12:51
Materiales: [ElProblemaControl.pdf]		[YouTube ▶]

En este vídeo se recapitulan, sin fórmulas, las ideas básicas de las etapas iniciales en el control de un sistema multivariable/complejo: se recuerda la necesidad de elegir referencias, sensores y actuadores, de elegir punto de operación, estructura de control, sintonizar reguladores y hacer pruebas y modificaciones según el resultado de las pruebas experimentales. Básicamente, resume las ideas intuitivas detrás de los razonamientos de los vídeos [424] y [427].

Luego, se detalla la importancia de elegir referencias primarias y secundarias que puedan ser movidas con gran amplitud y buen condicionamiento por los actuadores, cuyo punto de funcionamiento óptimo no cambie con las perturbaciones y que deberían ser medidas y controladas todas las variables (internas, estado) que estén asociadas a un almacenamiento de materia o energía importante.

[465: ect1b]	Estructuras de control: centralizado, descentralizado, jerárquico.	**  18:46
Materiales: [ElProblemaControl.pdf]		[YouTube ▶]

La primera parte de este vídeo describe qué objetivos de control podrían fijarse en un problema complejo... objetivos estáticos/dinámicos de seguimiento de referencia, de rechazo de perturbaciones, de robustez ante error de modelado, etc.

Luego, hay que decidir el flujo de información (estructura de control). En control monovariante, sólo tenemos que decidir si bucle *abierto* o bucle *cerrado*, en principio (realmente en muchos casos se debe abrir el bucle a alta frecuencia, filtrando ruido de medida, pero esos refinamientos no son objetivo de este vídeo).

En problemas multivariante, la selección de estructura de control implica que hay que elegir entre un extremo (centralizado, todos los sensores van a un computador que hace cálculos complejos y devuelve todos los actuadores) y otro (descomponer un problema en subproblemas, idealmente llegando a subproblemas de control puramente monovariante y reguladores PID). Cada extremo tiene sus ventajas e inconvenientes, y en la práctica las soluciones intermedias jerárquicas intentan extraer lo mejor de ambos enfoques. Este vídeo discute las diferentes ideas implicadas en todo ello.

Capítulo 14

Control multivariable: estructura descentralizada

14.1 Control multibucle

[466: ectmb] Control multibucle (totalmente descentralizado):
emparejamiento con metodología ganancia relativa (RGA)
Materiales: [CtrlDescen.pdf]

***  19:24
[YouTube ►]


Este vídeo discute las siguientes ideas:

- Control multibucle totalmente descentralizado: planteamiento del problema en matriz de transferencia (introducción al concepto de matriz de transferencia en vídeo [131]), diagrama de bloques. Emparejamientos posibles.
- Matriz de ganancia relativa (*relative gain array*, RGA). Uso para el problema del emparejamiento.
- Ejemplo numérico sobre un sistema de 3 entradas y 3 salidas.

Nota: algunos de los emparejamientos posibles según la metodología RGA (elementos asociados > 0) podrían ser descartados por un criterio complementario (criterio de Niederlinski) cuyos detalles se discuten en el vídeo [470].

Un ejemplo de aplicación a un problema de control de concentración y caudal en un sistema de tanques se discute en el vídeo [533].

[467: fdtapml] Control multibucle: función de transferencia aparente (ejemplo
Matlab)
Materiales: [[Cód.:](#) RGAmezclaFdTAparent.zip] [[PDF](#)]

****  09:50
[YouTube ►]

Este vídeo ilustra el concepto de función de transferencia “aparente” entre una entrada y una salida cuando se cierran otros bucles, que motiva la metodología de ganancia relativa para selección de emparejamiento discutida en el vídeo [466].

Se pone como ejemplo el modelo de tanque de mezclado, modelado en el vídeo [22], cuya linealización se discute en el vídeo [95].

Para ese proceso, se calculará la matriz de ganancia relativa (RGA) y se comprobará la función de transferencia aparentes entre $q_{a,in}$ y x_a cuando se cierra el bucle $q_{b,in} \rightarrow h$ y cómo la RGA predice el cambio de ganancia.

Del mismo modo, también se calculará y simulará la función de transferencia aparentes entre $q_{b,in}$ y x_a cuando se cierra el bucle $q_{a,in} \rightarrow h$.

[468: mbmix] Caso de estudio metodología ganancia relativa: control multibucle de un tanque de mezclado (Matlab)

Materiales: [[Cód.: RGAmezclaPruebasMultibucle.zip](#)] [[PDF](#)]

*** 10:53

[YouTube ▶]

Este vídeo ilustra el funcionamiento de la metodología de ganancia relativa para selección de emparejamiento discutida en el vídeo [466].

Se pone como ejemplo el modelo de tanque de mezclado, modelado en el vídeo [22], cuya linealización se discute en el vídeo [95]. Al modelo se le ha añadido un retardo en la salida de la concentración para aproximar dinámica del sensor y tiempo de mezclado. *Nota:* en aplicaciones reales, dependiendo de la velocidad de la agitación y de la viscosidad de los reactivos, dicha dinámica puede tener un retardo o constantes de tiempo que influyan durante una parte significativa del transitorio, y que dificulten el control preciso de la concentración.

Nota: si se desea una aproximación más sencilla a un proceso de mezclado, el emparejamiento RGA en un mezclado “estático” (instantáneo, sin dinámica) se discute en el vídeo [533]; se recomienda la visualización previa de ese vídeo para una mejor comprensión de los procesos de mezclado.

El concepto de “función de transferencia” aparente subyacente a la metodología se aborda en el vídeo [467], pero no es necesario para comprender los desarrollos aquí.

Para ese proceso, se calculará la matriz de ganancia relativa (RGA) y se harán pruebas sobre dos emparejamientos:

- El “malo”: $q_a \rightarrow x_a, q_b \rightarrow h$.
- El “bueno”: $q_a \rightarrow h, q_b \rightarrow x_a$.

Se diseñarán reguladores PD para el caso donde el resto de los bucles no esté conectado (o sea, con la función de transferencia “aislada” correspondiente). Se observará que el cambio en prestaciones cuando los dos bucles están conectados es mucho menor cuando se usa el emparejamiento recomendado por la RGA.

La simulación de estos bucles con dinámica del actuador y saturación se aborda en el vídeo [469], usando Simulink, donde se implementarán también de forma muy sencilla las estrategias *antiwindup* que completarían el diseño del control.

[469: mbmixaw] Control multibucle de un tanque de mezclado: dinámica actuador, saturación y antiwindup (Simulink)

Materiales: [[Cód.: SimulinkMultibucleTanqueMezcla.zip](#)] [[PDF](#)]

*** 09:27

[YouTube ▶]

Este vídeo simula utilizando Simulink los controladores diseñados y analizados en el vídeo [468].

Para ese proceso, se harán pruebas sobre dos emparejamientos:

- El “malo”: $q_a \rightarrow x_a, q_b \rightarrow h$.
- El “bueno”: $q_a \rightarrow h, q_b \rightarrow x_a$.

La diferencia con el referido vídeo es que aquí se añadirá una dinámica de actuador $G_{act}(s) = 1/(0.02s + 1)$ y una saturación a ± 2 de ambos caudales. Esta dinámica y saturación empeorarán el comportamiento del emparejamiento no recomendado, y causarán el efecto *wind-up* en la acción integral en el bucle con el emparejamiento recomendado por la RGA.

Por ello, se propone usar el “tracking input” del bloque PID de Simulink para implementar un antiwindup realimentando la salida “real” del actuador tras la saturación (los detalles teóricos detrás de esta metodología de antiwindup se abordan en los vídeos [521], [522] y [523]). Se comprueba que la respuesta es satisfactoria y ese sería, por tanto, el diseño final

recomendado en este caso de estudio multibucle. Si se considerara que el acoplamiento (transitorio en una variable cuando se cambia la otra referencia) fuera no satisfactorio, entonces deberían considerarse otras opciones como el desacoplamiento (abordado en el vídeo [487]).

[470: niede]

Criterio de emparejamiento de Niederlinski, relación con RGA y ejemplo Matlab

Materiales: [CÓD.: NiederlinskiIntro.mlx] [PDF]

*** 10:41
[YouTube ▶]

En este vídeo se presenta el criterio de emparejamiento de Niederlinski, que, para plantas estables en bucle abierto dice que deberían preferirse los emparejamientos donde $\det(G)/\det(\text{diag}(G)) > 0$.

Este criterio es una alternativa complementaria al de ganancia relativa (vídeo [466]). En el caso de 2 entradas y 2 salidas proporcionan la misma información, pero no así en plantas de mayor dimensión. Un ejemplo Matlab en una planta 4×4 ilustra la metodología combinada Niederlinski+RGA.

La demostración del criterio se discute en el vídeo [471].

[471: niedep]

Criterio de emparejamiento de Niederlinski: demostración

Materiales: [NiederlinskiStableProof.pdf]

**** 12:35
[YouTube ▶]

Este vídeo presenta la demostración del criterio de emparejamiento de Niederlinski, cuya utilización complementaria con la RGA se discutió en el vídeo [470], pero del que también se incluye la demostración en el caso estable para el lector interesado.

La demostración basa en la idea de que un polinomio $s^n + \dots + \gamma_1 s + \gamma_0$ debe tener todos los coeficientes no negativos como condición necesaria para que sus raíces sean estables (parte real negativa), con lo que puede deducirse una condición del signo de la ganancia a baja frecuencia de las acciones integrales de los reguladores de un bucle.

14.1.1 Consideraciones adicionales sobre la matriz RGA

[472: rga2]

Matriz de ganancia relativa (RGA): propiedades adicionales

Materiales: [RGAampli.pdf]

*** 11:00
[YouTube ▶]

En este vídeo se discuten propiedades adicionales de la matriz de ganancia relativa cuya justificación en selección de emparejamientos para control multibucle se presentó en el vídeo [466].

En concreto, se demuestra su independencia del escalado (adimensional), que la suma de filas y columnas da 1, y se enuncia el hecho de que en plantas triangulares (o permutaciones de ella) la RGA produce el resultado intuitivamente esperado (es la identidad o permutación adecuada) dado que sólo hay un emparejamiento posible.

En el minuto [04:30] se discute la RGA de sistemas con *integradores*. Al ser la ganancia estática infinita se debe sustituir por el límite cuando $s \rightarrow 0$, evaluado numéricamente dando valores pequeños a s .

La parte final del vídeo plantea la relación de la RGA con el número de condición mínimo. Primero se revisa qué es el número de condición de una matriz (detalles en vídeo [786]), y el concepto de condicionamiento óptimo antes escalado diagonal

$$\gamma_{MIN}(G) := \min_{E_y, E_u \text{ diag.}} \gamma(E_y^{-1} G E_u),$$

ver detalles en vídeo [893]. Tras esta revisión, se presenta una fórmula que da una cota inferior del número de condición de una matriz en función del valor absoluto de los elementos de la RGA: la presencia de elementos grandes en la RGA indica problemas sustanciales de condicionamiento que no pueden ser solucionados por un cambio de unidades trivial y que aconsejarían cambiar la selección de variables controladas/manipuladas o la posición/tecnología de sensores y actuadores.

[473: rgacondm] Relación RGA, escalado y condicionamiento: ejemplo Matlab

*** 07:23

Materiales: [Cód.: rgamincond.mlx] [PDF]

[YouTube ▶]

Este vídeo presenta un ejemplo Matlab ilustrando la independencia del escalado y la relación que existe entre los elementos de la RGA y el número de condición óptimo tras escalado diagonal en un proceso. Estos conceptos han sido abordados desde un punto de vista teórico en el vídeo [472].

Primero, se presenta una matriz mal condicionada pero que un escalado de las entradas transforma en una muy bien condicionada (en ésta, todos los elementos de la RGA son menores a 1). En la segunda parte, se presenta otra matriz que resulta en elementos grandes de la RGA que indican que no existe ningún escalado diagonal que pueda conseguir un condicionamiento menor de 38 y, por tanto, controlar un proceso con dicha matriz de ganancia estática seguramente resulte muy difícil (sensible a errores de modelado).

14.2 Prealimentación (Feedforward)

[474: ffintro] Prealimentación (feedforward): conceptos básicos

** 11:00

Materiales: [FF1.pdf]

[YouTube ▶]

En control avanzado se utilizan sensores adicionales: si las variables medidas pueden controlarse con los actuadores disponibles, entonces se trata de control *en cascada*; si son perturbaciones (variables “exógenas” no controlables por mis actuadores) entonces se trata de control feedforward o por **prealimentación**.

En este vídeo se presenta, de forma intuitiva, el significado de dicha prealimentación, mediante ejemplos sin cálculos o con cálculos muy sencillos, para darse cuenta de que la idea es bastante frecuente. La parte final del vídeo plantea el problema en términos formales de modelos y sistemas de ecuaciones con los mismos. El caso de modelos lineales se discutirá en detalle en los vídeos [475]y [477].

[475: fflin] Prealimentación (disturbance feedforward) en modelos lineales

*** 10:43

Materiales: [FFLinSISO.pdf]

[YouTube ▶]

En muchos casos en control están disponibles medidas de perturbaciones, dando lugar al control feedforward cuya interpretación “intuitiva” se discute en el vídeo [474].

En este vídeo se discute el caso de un modelo lineal $y = Gu + Hd + Ww$ donde d es medible, w no lo es, y u es la variable manipulada (actuador). Se demuestra que $u_{FF} = -G^{-1}Hd$ cancela el efecto de d sobre y . En muchas aplicaciones, en vez de $G^{-1}(s)H(s)$ se usa simplemente la ganancia estática (aunque, dependiendo de la presencia o no de controladores con acción integral y los objetivos de control, podría no ser la mejor opción).

Un ejemplo de identificación “experimental” (simulación, claro, en realidad) de este tipo de modelos G y H de perturbación lineales aparece en el vídeo [479]. La simulación de los diseños de prealimentación estáticos se aborda en el vídeo [480], y el caso dinámico en el [481].

La parte final del vídeo discute un ejemplo de este tipo de estructura de control.

[476: fmal]

Prealimentacion multivariable: ideas preliminares (sencillas pero parcialmente incorrectas)

***  08:57

Materiales: [[CÓD.: PruebasFFMimoMal.mlx](#)] [[PDF](#)]

[YouTube ▶]

Este vídeo discute ideas preliminares sobre cómo abordar el caso de prealimentación multivariable.


La idea más sencilla sería, si hay un control multibucle, el utilizar para cancelar una perturbación sobre cierta salida el mismo actuador que el regulador de dicha salida... en efecto, ese cálculo puede hacerse mediante un “cociente” entre dos ganancias (o dos FdT en caso dinámico) y explicarse de forma muy directa, aplicando las ideas monovariantes del vídeo [475].

Esa idea funciona razonablemente bien en un proceso “diagonal dominante” (que el elemento más grande de una columna sea correspondiente el emparejamiento multibucle elegido, y éste sea mucho más grande que el resto), como se ve en un primer ejemplo Matlab. Si el proceso no es diagonal dominante, cancelar la perturbación en un canal puede producir un efecto en los otros más grande del que se quería evitar (debido al acoplamiento: efecto del actuador utilizado sobre otras variables controladas). En ese caso, deberían utilizarse fórmulas con inversas de matrices, cuya teoría se discute en el vídeo [477].

El enfoque matricial exacto es el recomendable en general desde un punto de vista teórico. Los cocientes monovariantes diagonales justificarían una explicación “simple” pero *incorrecta* en entornos con interlocutores donde el concepto de *matriz* y su inversa no les fuera familiar... sólo sería correcta en el caso de utilizar desacoplamiento (segunda parte del vídeo [477]) junto al feedforward. Un ejemplo numérico de estas ideas se aborda en el vídeo [525].

[477: ffdec]

Prealimentación en modelos multivariantes y relación con desacoplamiento

****  08:55

Materiales: [[FFLinMIMO.pdf](#)]

[YouTube ▶]

Este vídeo discute la prealimentación (feedforward) con modelos multivariantes. El hecho es que las fórmulas son las mismas que las discutidas en el vídeo [475], simplemente dándose cuenta de que los elementos que intervienen son *matrices* de transferencia, en vez de *funciones* de transferencia monovariantes (ver vídeo [131] para una introducción al concepto de matriz de transferencia). En muchas aplicaciones, se hace feedforward sólo con matrices de ganancia estática.


Si el proceso tiene integradores, habría que evaluar las ganancias como un límite cuando la frecuencia tiende a cero, ver un ejemplo en vídeo [545].

La segunda parte del vídeo discute la relación entre prealimentación y desacoplamiento (cambio de variable $u = D\nu$ que transforme la matriz de transferencia o de ganancia aparente ante las nuevas entradas ν de forma que $y = (GD) \cdot \nu$ parezca diagonal, detalles en vídeos [483] y [484]). En principio, si se supone que va a haber un desacoplamiento preexistente, entonces el feedforward se puede calcular considerando únicamente la diagonal de las matrices (dado que los términos no diagonales serán cancelados por el desacoplamiento). El desarrollo teórico comprueba que, desde un punto de vista formal, produce el mismo resultado que la prealimentación sin desacoplamiento preexistente discutida en la primera mitad del vídeo. Un ejercicio numérico aparece en el vídeo [525].

14.2.1 Ejemplos y casos de estudio

[478: ffret]

Prealimentacion con retardo en efecto de perturbación: ejemplo Matlab

****  11:40

Materiales: [[CÓD.: FFcasoestudio2.mlx](#)] [[PDF](#)]

[YouTube ▶]

Este video presenta un caso de estudio en Matlab donde se diseña una prealimentación (feedforward) para un modelo $y = \frac{2}{(s+1)^2} \cdot u + \frac{1}{2s+1} e^{-1.2s} \cdot d$.

Se observa que existe un retardo en el efecto de la perturbación, de modo que si d es medible, se sabrá de su presencia 1.2 segundos antes de que empiece a manifestarse su efecto en y .

Se compararán tres opciones, tanto en respuesta temporal (step) como en frecuencia (bode):

- Prealimentación estática, cociente de ganancias
- Prealimentación dinámica $G^{-1}G_{pert}$ añadiendo un polo rápido para realizabilidad, pero manteniendo el retardo de 1.2 segundos en la actuación u .
- Prealimentación dinámica, anticipándose 0.4 segundos en la actuación sobre u , usando $e^{-ds} = 1/e^{ds} \approx 1/(1+ds)$.

[479: idff] **Perturbaciones medibles: identificación para feedforward**
(Matlab, procest)

Materiales: [[CÓD.:](#) modeladoparaFF.zip] [[PDF](#)]

*** 13:23

[YouTube ▶]

Este vídeo simula en Simulink un proceso (desconocido) donde existen dos sensores, uno de variable controlada y otro un sensor de una perturbación (medible). Si se dispone de un modelo adecuado del efecto de dicha perturbación, podría hacerse prealimentación (feedforward).

Con los datos de la simulación, guardados mediante un bloque **ToWorkspace**, se determinan los puntos de funcionamiento como la media de ciertas regiones de los datos experimentales, y se utiliza **procest** con los incrementos para estimar las funciones de transferencia G y G_{pert} en un modelo $y = Gu + G_{pert}d$. Se comprueba el ajuste con distintos parámetros de procest (número de polos, presencia o no de retardo); se escoge finalmente modelos de primer orden + retardo.

Con el modelo identificado, se abordará la prealimentación estática en el vídeo [480] y la prealimentación dinámica en el vídeo [481].

[480: ffex1] **Prealimentación estática: ejemplo Matlab/Simulink**
perturbaciones medibles

Materiales: [[CÓD.:](#) PrelimFF123.zip] [[PDF](#)]

*** 07:10

[YouTube ▶]

Este vídeo plantea cómo hacer prealimentación (estática) en un proceso donde tanto la función de transferencia entre variable manipulada (u) y controlada (y), como aquella entre una perturbación (p) medible y la salida y son modelos de primer orden + retardo. Los detalles de la identificación se abordan en el vídeo [479], pero lo más importante de estos aspectos preliminares se resume hasta el instante [01:46] de este vídeo.

El resultado es un modelo $y = Gu + G_{pert}d$.

En este vídeo se plantea la prealimentación estática en ese modelo, que básicamente es $u_{FF} = -dcgain(G^{-1}) \cdot dcgain(G_{pert}) \cdot d$. Ello cancela el efecto de d una vez los transitorios de G_{pert} y de G han terminado.

A partir del instante [03:15] se implementa en Simulink dicho controlador prealimentado. En concreto, se integran los puntos de funcionamiento de la perturbación en el diagrama, dado que el modelo linealizado lo es en variables *incrementales*. Añadiendo un bloque de ganancia, se implementa la prealimentación estática y se compara el resultado con/sin prealimentación (feedforward) mediante el Data Inspector de Simulink, que permite comparar diferentes simulaciones superponiendo resultados.


Por su sencillez, la prealimentación estática es la primera y más usada opción en aplicaciones.

De hecho, es importante resaltar que la acción integral también cancela errores de posición en régimen permanente (cancela efecto a baja frecuencia de perturbaciones) por lo que no está garantizado que en un caso general la prealimentación estática vaya a mejorar las prestaciones respecto a un regulador en bucle cerrado con acción integral bien sintonizado.

Si la prealimentación estática no funciona adecuadamente, podría plantearse la prealimentación dinámica; en este proceso, la opción dinámica se abordará en el vídeo [481].

[481: ffex2] **Feedforward (prealimentación) dinámico (1er ord.+retardo):**
ejemplo Matlab

Materiales: [[CÓD.:](#) PrealimFF123.zip] [[PDF](#)]

****  14:41

[YouTube ▶]

Este vídeo plantea cómo hacer prealimentación (dinámica) en un proceso donde tanto la función de transferencia entre variable manipulada (u) y controlada (y), como aquella entre una perturbación (p) medible y la salida y son modelos de primer orden + retardo. Los detalles de la identificación se abordan en el vídeo [479], y la prealimentación estática y su simulación se discuten en el vídeo [480], cuya visualización previa se recomienda; de todos modos, la idea estática se resume hasta el instante [01:30] de este vídeo.

Aquí se plantea la prealimentación dinámica. No obstante, en este caso existen *graves dificultades*: el retardo de la perturbación es de 0.52 segundos, mientras que el retardo de la acción de la variable manipulada es de 4.25 segundos, con lo que hay 3.75 segundos donde el efecto de la perturbación es imposible de ser cancelado (la prealimentación ideal $F(s) = -G^{-1}(s)G_{pert}(s)$ es no causal).

Desde un punto de vista teórico, se plantea comparar en tiempo y frecuencia dos opciones aproximadas dinámicas (además de la estática):

1. Aproximar el retardo $e^{-ds} \approx 1$, válido a frecuencias muy bajas, o sea, hacer prealimentación dinámica como si el retardo no existiera.
2. Aproximar el retardo por un polo $e^{-ds} = 1/e^{ds} = 1/(1 + ds)$, que también es válida a baja frecuencia.

Primero se simulan en Matlab, tanto la respuesta de la acción de control $-u(s) = F(s)d(s)$, donde $F(s)$ es una constante o funciones de transferencia en los dos casos dinámicos estudiados –, como el error residual $e(s) = (G(s)F(s) + G_{pert}(s)) \cdot d(s)$. En esa simulación se analizan las respuestas en tiempo y en frecuencia. Se observa un efecto “cama de agua” (*waterbed effect*, en la literatura en lengua inglesa): cuanto mejor se cancelan las perturbaciones de baja frecuencia, más se amplifican (respecto a no hacer nada) a altas frecuencias.

Dado que la validez de las aproximaciones estáticas o dinámicas sólo es a baja frecuencia, habría que ver aplicación por aplicación si el efecto de esa prealimentación realmente mejora sustancialmente la respuesta comparada con un regulador en bucle cerrado (realimentación) con acción integral adecuadamente sintonizado.

Nota: utilizando la aproximación de Padé y control óptimo \mathcal{H}_∞ , un problema muy similar al aquí discutido se aborda en los vídeos [660] (sin retardo) y [661] (con retardo).

[482: ffex3] **Feedforward (prealimentación) dinámico (1er ord.+retardo):**
implementación Simulink

Materiales: [[CÓD.:](#) PrealimFF123.zip] [[PDF](#)]

***  09:34

[YouTube ▶]

Este vídeo plantea cómo hacer prealimentación (dinámica) en un proceso donde tanto la función de transferencia entre variable manipulada (u) y controlada (y), como aquella entre una perturbación (p) medible y la salida y son modelos de primer orden + retardo. En concreto se aborda la implementación en Simulink de las opciones de prealimentación estática (vídeo [480]), y las opciones de prealimentación dinámica aproximada consideradas en el vídeo [481]. Las conclusiones de este último vídeo se resumen en los dos primeros minutos de este vídeo.

Primero, se simula un escenario sin feedforward donde las entradas de perturbación son escalones. Luego, se ejecutan cada una de las tres opciones (estática, aproximación $e^{-ds} \approx 1$ y aproximación $e^{-ds} \approx 1/(1 + ds)$). Con el Data Inspector de Simulink se examinan y comparan las salidas y acciones de control de cada una de las opciones. Los efectos de saturación y errores de modelado (el proceso “real” dentro de Simulink no es primer orden más retardo) hacen que apenas haya diferencias entre las tres opciones de feedforward.

A partir del minuto [06:20] se introduce una perturbación senoidal (nótese que, en la práctica, dado que las perturbaciones no son manipuladas por el usuario, sería un experimento seguramente imposible de realizar: las perturbaciones toman los valores que el azar u otros agentes decidan que tomen), con una frecuencia más alta de las que son bien canceladas, con lo que ya el vídeo [481] sugería que a frecuencias altas podría ser “peor el remedio que la enfermedad”. Debido a saturaciones y errores de modelado, y a que teóricamente ya el funcionamiento previsto no era bueno, en efecto se observa que ninguna de las opciones de prealimentación reduce el efecto de la perturbación sobre la variable controlada a esa frecuencia.

14.3 Desacoplamiento

[483: ectdc]

Estructuras de control multivariable: Control por desacoplamiento (generalidades)
Materiales: [CtrlDescen.pdf]

*** / **** 19:49
 [YouTube ▶]

Este vídeo describe las ideas básicas del desacoplamiento feedforward:

- diagrama de bloques (feedforward, desacoplamiento en bucle abierto).
- Requisitos para su diseño: realizabilidad y cancelaciones polo/cero.
- Desacoplamiento estático (ganancia diagonal) versus dinámico (FdT diagonal a todas las frecuencias)
- Ventajas e inconvenientes en la práctica.
- Introducción al desacoplamiento SVD (basado en valores singulares –vídeo [931]–, válido incluso en plantas *no cuadradas*). Ver vídeos [489], [496] y [497] para detalles de implementación y ejemplos de código. ****

Notas:

1.- Las fórmulas de desacoplamiento *feedforward* $D = G^{-1}M$ aquí consideradas también sirven para diseñar bucles cerrados, resultarían idénticas si se considerara la opción de control por modelo interno (vídeos [391], [393], ..., pero ahora entendiendo como matrices todos los elementos de dichos materiales... el desacoplador D aquí y el parámetro de Youla Q en IMC son conceptos muy relacionados).

2.- Existen dos alternativas al desacoplamiento feedforward: (a) el desacoplamiento en *forma inversa*, discutido en el vídeo [484], así como (b) lo que se denomina desacoplamiento por *realimentación*, del que un ejemplo aparece en el vídeo [502].

[484: desacinv]

Desacoplamiento inverso (inverted decoupling): justificación teórica
Materiales: [DesacPrelim1.pdf]

*** 10:56
 [YouTube ▶]

Este vídeo discute el desacoplamiento en forma inversa, donde se diseña una matriz $F = \text{inv}(\text{diag}(G)) * (G - \text{diag}(G))$, de modo que el desacoplador es $D = (I + F)^{-1}$. Se comprueba que $GD = \text{diag}(G)$ y se discute su justificación intuitiva en base a fórmulas de prealimentación (cancelación de perturbaciones medibles) dado que F resulta de considerar al resto de entradas como “perturbaciones” al bucle que se intenta hacer diagonal.

[485: desacinvM]

Desacoplamiento inverso (inverted decoupling) en
implementación matricial: ventajas e inconvenientes

Materiales: [DesacPrealim2a.pdf]

*** 09:35

[YouTube ▶]

En este vídeo se presentan las dos posibles implementaciones del desacoplamiento inverso cuya derivación se discutió en el vídeo [484].

En concreto, se analiza en detalle la implementación directa matricial del desacoplador inverso $D = (I + (\tilde{G})^{-1} \cdot (G - \tilde{G}))^{-1}$

Básicamente, su implementación como una matriz (de transferencia o constante) tiene la ventaja de que evita determinados problemas debidos a la realimentación (bucle algebraico) de la otra alternativa de implementación (vídeo [486]), aparte de conservar la diagonal de modo que si se tenía un diseño inicial multibucle, este diseño no necesitaría ser modificado (al menos a frecuencias bajas –reguladores lentos– si sólo se usa la matriz de ganancia).

No obstante, plantea problemas al necesitar modificar bucles incorporando elementos de cálculo matricial (no estándar en entornos con sólo PIDs), así como las cuestiones de cancelación de polos inestables o ceros de fase no mínima. El otro inconveniente importante es que la función de transferencia aparente cambia si se satura alguno de los actuadores, y que resulta muy difícil hacer el *antiwindup* a los PIDs. En esencia, estos son los inconvenientes de todo desacoplamiento implementado como una matriz D tal que GD sea diagonal.

[486: desacinvF]

Desacoplamiento inverso en implementación con bloques SISO:
ventajas e inconvenientes

Materiales: [DesacPrealim2b.pdf]

**** 11:04

[YouTube ▶]

En este material se discuten las ventajas e inconvenientes de la implementación con bloques de una entrada y una salida del desacoplamiento en forma inversa $(I + F)^{-1}$ considerado en el vídeo [484].

Comparando con la implementación como una matriz (vídeo [485]), las ventajas más claras son la evitación del uso de elementos de proceso que hagan cálculo matricial (o que simulen matrices de transferencia), así como su especial adaptación a esquemas de control en cascada y buena resiliencia a saturación o cambio automático/manual sin que el resto de bucles vean cambiada su función de transferencia aparente.

A cambio, existe la desventaja de que se genera un bucle realimentado adicional (bucle algebraico si se hace con matriz de ganancia o si los elementos tienen transmisión entrada→salida a frecuencias altas) que puede introducir dinámicas espúreas cuya estabilidad o influencia en el bucle completo debería ser comprobada. La dinámica del desacoplador es más rápida y da menos problemas si F es pequeña (proceso diagonal-dominante).

14.3.1 Caso de estudio: tanque de mezclado

[487: mixdecaw] Tanque de mezclado, caso de estudio: control en cascada + desacoplamiento inverso + antiwindup

Materiales: [CasoEstudioMezcladoCascDecAW.pdf]

**** 11:40

[YouTube ▶]

Este vídeo presenta un diseño “intuitivo” de un control inicial multibucle para un tanque de mezclado (ver detalles en vídeo [468]), al que se le incorpora un control en cascada de unas servoválvulas con sensor extra de caudal en cada una de ellas.

Una vez planteado ese diseño original, si la interacción entre los bucles no fuera satisfactoria, podría incluirse un desacoplamiento. Dado que el desacoplamiento en forma inversa está particularmente indicado si se dispone de sensor de las acciones de control, como es el caso aquí, se elegirá esa forma de implementar el desacoplamiento, con bloques SISO (los detalles teóricos aparecen en el vídeo [484], y las ventajas e inconvenientes en aplicaciones en el vídeo [486]).

El desacoplamiento en forma inversa también resulta especialmente conveniente si se desea implementar *antiwindup* cuando los actuadores de los bucles esclavos saturan (Nota: el antiwindup en el caso multibucle también se discute en el vídeo [469] que podría ser de interés visualizar antes de este). Se propone un PID con “tracking input” (detalles teóricos en vídeo [521]) y un diagrama de bloques específico para desacoplamiento y feedforward necesario para llevar a cabo ese antiwindup (que no era necesario en el multibucle del vídeo [469]), con lo que se completaría el diseño propuesto.

La simulación con Simulink de la estructura propuesta se presenta en el vídeo [488].

[488: mixdecawS] Simulación de Tanque de mezclado: multibucle y desacoplamiento inverso + antiwindup (Simulink)

Materiales: [Cód.: SimDesacAntiWindup.zip] [PDF]

**** 06:17

[YouTube ▶]

Este vídeo simula en Simulink el diseño considerado en el vídeo [487], comparando la solución multibucle recomendada por la RGA con el desacoplamiento en forma inversa; se ha considerado una dinámica de actuadores (que podría ser originada, por ejemplo, como la dinámica en bucle cerrado de un control en cascada). El vídeo es bastante breve, dado que los conceptos teórico-prácticos subyacentes se han desarrollado en el vídeo [487] y la simulación del control multibucle (con dos posibles emparejamientos y antiwindup) se ha discutido también con bastante detalle en el vídeo [469].

14.3.2 Desacoplamiento SVD

[489: dsvd1] Desacoplamiento SVD (1): ideas básicas y diagrama de bloques

Materiales: [DesacSVD2.pdf]

**** 12:18

[YouTube ▶]

Este vídeo discute en detalle los cambios de variable de escalado y paso a componentes principales del desacoplamiento SVD, que se había esbozado en los minutos finales del vídeo [483].

Aquí se lleva a diseño de control la idea inicial de un cambio de variable $G_{desacsvd} = U^T \cdot G \cdot V$, propuesto en la parte final del vídeo [446].

Presenta el diagrama de bloques de implementación, y el concepto de referencia realizable (proyección ortogonal sobre los componentes que se desea controlar).

El vídeo [489] explica también estos conceptos (como base de un caso de estudio en una serie de vídeos); se recomienda visualizarlo conjuntamente con éste.

14.3.3 Explicación más detallada, comprensión “intuitiva”

[490: dsvdintu1] Desacoplamiento SVD, caso estudio (1): revisión teoría

Materiales: [SVDdecouplingTSTteoriaESP.pdf]

*** 16:44

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo realiza una revisión teórica del desacoplamiento SVD, en la línea del vídeo [489], reformulando/complementando la exposición allí presentada.

El vídeo revisa el concepto de maniobras principales, direcciones de entrada, de salida y espacio nulo. Luego, propone cambio de variable de modo que $y^{SVD} = U^T y$, $u^{SVD} = V^T u$. Con ese cambio de variable $y^{SVD} = S u^{SVD}$ si $y = (USV^T)u$ es la descomposición SVD de la matriz de ganancia estática. De este modo, el comportamiento aparente es diagonal S , y se pueden diseñar reguladores (lentos, dado que el desacoplamiento es a frecuencia cero) explotando esa idea, resultando $u = VK(s)U^T e$, con $K(s)$ diagonal controlando por separado cada maniobra principal.

Si algunas de las maniobras principales de menor ganancia no son controladas (por culpa de poca ganancia o/y mal condicionamiento numérico si se hiciera), para evitar saturación frecuente y sensibilidad a errores de modelado, entonces $u = VK(s)U^T e$ se construiría sólo con un subconjunto de columnas de U y V (relacionado con la llamada ‘descomposición SVD económica’).

[491: dsvdintu2] Desacoplamiento SVD, caso estudio (2): 1 vble. controlada, 2 manipuladas

Materiales: [[CÓD.:](#) SVDdecouplingTSTESP.mlx] [[PDF](#)]

**** 16:40

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este video discute el concepto de maniobras principales y desacoplamiento SVD en un proceso de 1 salida controlada y 2 entradas. Para entender el significado “intuitivo” de lo que se propone, se ilustra con un ejemplo físico de calentar una pieza (un termopar da la lectura de la variable a controlar) con una resistencia y un ventilador.

Obviamente, existen infinitas soluciones a $y = Gu$, y con los escalados adecuados, querríamos usar aquella que minimizara $u_1^2 + u_2^2$.

En concreto, la descomposición en valores singulares $G = USV^T$ de una matriz que es un vector fila tiene $U = 1$, $S = \text{norm}(G)$, y $V = G^T / \text{norm}(G)$, esto es, un escalado a norma 1 de la matriz original (y las direcciones ortogonales a G forman el espacio nulo). La pseudoinversa es $G^T / \text{norm}(G)^2$. Si la teoría no la tienes fresca, revisa el vídeo [490], previo a éste.

Nada es especialmente interesante del SVD, en el sentido que realmente las direcciones de entrada con efecto no nulo son simplemente proporcionales a la ganancia de cada actuador.

Pero el objetivo es captar la intuición detrás de que las soluciones de mínimos cuadrados reparten el esfuerzo de control de forma proporcional a la ganancia y, por tanto, podríamos pensar en que la única variable controlada se controlara mediante un actuador “virtual” que moviera los actuadores físicos de forma coordinada repitiendo la misma orden multiplicada por la ganancia.

El vídeo [492] continúa con el caso “dual” de varias salidas a controlar con una única entrada manipulada.

[492: dsvdintu3] Desacoplamiento SVD, caso estudio (3): 3 vble. controladas, 1 manipulada

Materiales: [[CÓD.:](#) SVDdecouplingTSTESP.mlx] [[PDF](#)]

*** 13:45

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo continua con el caso de estudio sobre la intuición detrás de maniobras principales y desacoplamiento SVD. En concreto, presenta un caso con 3 sensores (temperaturas de una pieza) a controlar con un único actuador (resistencia). La matriz de ganancia estática es un vector columna, y obviamente no pueden controlarse las tres temperaturas a referencias arbitrarias, con lo que buscaremos ajustarlas por “mínimos cuadrados”.

De hecho, dependiendo de tu familiaridad con estos temas, podrías visionar el vídeo [584] donde se discuten los mínimos cuadrados, pseudoinversas, etc. sin mención al SVD aplicados a un horno prácticamente idéntico al del caso estudio de este vídeo y siguientes.

En este caso el SVD de $G = USV^T$ resulta en $V = 1$ (no hay direcciones de entrada, ya que es única), y U proporcional a G , simplemente siendo $G/norm(G)$. Es el caso “dual” al visto en el vídeo [491], donde G era vector fila.

En este caso, el significado de la única maniobra principal y de la pseudoinversa es que el ajuste por mínimos cuadrado se consigue controlando una variable “virtual”, que denominaremos y^{SVD} , que es una media ponderada de las temperaturas, ponderada en función de la ganancia estática.

La parte final del vídeo, para concluir la comparación entre el significado intuitiva de G como “fila” o como “columna” esboza rápidamente un ejemplo de 1 variable controlada y 3 manipuladas, totalmente análogo al del vídeo [491].

[493: dsvdintu4] Desacoplamiento SVD, caso estudio (4): referencia realizable

Materiales: [[CÓD.:](#) SVDdecouplingTSTESP.mlx] [[PDF](#)]

*** 13:20

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo es continuación del vídeo [492], revisado en los primeros tres minutos.

Aquí añadiremos puntos de operación de actuadores y ahondaremos en el concepto de ‘referencia realizable’ (el valor que haría cero $e^{SVD} = r^{SVD} - y^{SVD}$ si, por ejemplo, el PID del vídeo referido tuviera acción integral. Esta referencia será $r^* = Ur^{SVD} = UU^T r$.

Conviene indicar el valor de dicha referencia realizable al usuario, para que él sepa que la referencia que él ha dado no es realmente conseguible (no tenemos suficientes variables manipuladas), y cuál es la referencia real a la que el proceso tenderá.

Dado que con una única variable manipulada sólo puede controlarse una variable, también podríamos haber pensado en elegir un único termopar (el de más ganancia, por ejemplo) y poner un PID monovariante para él. Esta idea, obviamente de interés intuitivo, es discutida en los dos minutos y medio finales del vídeo. Claramente, podría ser la solución válida más sencilla... pero el objetivo de este caso de estudio es comprender las similitudes y diferencias con las otras soluciones de medias ponderadas de sensores (lo que sería seguramente diferente será el comportamiento ante perturbaciones).

[494: dsvdintu5] Desacoplamiento SVD, caso estudio (5): 3 vble. controladas, 4 manipuladas

Materiales: [[CÓD.:](#) SVDdecouplingTSTESP.mlx] [[PDF](#)]

**** 18:57

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute la interpretación intuitiva de las maniobras principales en un proceso de 3 salidas controladas y 4 entradas manipuladas: pieza con tres termopares controlada con dos resistencias y dos sopladores de aire frío.

El vídeo es continuación del caso de estudio en el que se analizó la interpretación intuitiva del SVD mediante un proceso de 1 variable controlada y varias manipuladas, o de varias controladas y una manipulada. Estos ejemplos anteriores se resumen en los primeros tres minutos de vídeo (detalles en vídeos [491] y [492]).

El objetivo es entender cómo las fusiones de actuadores o las medias ponderadas de sensores que aparecieron en los ejemplos anteriores pueden ser generalizados al caso completo multivariable.

La parte inicial del vídeo analiza las maniobras principales, ganancia y condicionamiento, concluyendo que sólo dos maniobras principales son razonablemente “controlables”.

A partir del minuto [12:30] se discute el diagrama de bloques que llevaría a cabo el desacoplamiento SVD, con un controlador separado (p. ej., PID) para cada una de las dos maniobras.

[495: dsvdintu6] Desacoplamiento SVD, caso estudio (6): subconjuntos/fusión manual de actuadores y sensores

Materiales: [[CÓD.:](#) SVDdecouplingTSTESP.mlx] [[PDF](#)]

**** 14:14

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo concluye el caso de estudio sobre la interpretación intuitiva del SVD y el desacoplamiento por valores singulares que comenzó con el repaso teórico en el vídeo [490], y con los ejemplos del [491] en adelante.

Aquí, se ahonda en el análisis del proceso de 3 variables controladas y 4 manipuladas del vídeo [494]. En concreto, dado que sólo se controlaban dos cosas, se propone:

- Despreciar el termopar central, controlando sólo dos sensores “físicos” en vez de “virtuales” (medias ponderadas de sensores físicos).
- Fusionar en cada lado del horno la resistencia y el soplador como un único actuador antes de analizar SVD, etc.

El resultado es un proceso “reducido” de 2 entradas y 2 salidas con buen condicionamiento, buena RGA, con lo que podría proponerse controlarlo simplemente con dos PIDs de ‘toda la vida’. La solución podría ser buena; relacionarla con las otras opciones basadas en SVD del vídeo [494] es el objetivo del vídeo, para que el SVD pueda darnos pistas sobre este tipo de posibilidades en procesos donde no comprendamos su tecnología tan claramente como en este ejemplo sencillo de termopares, resistencias, ventiladores.

Esto finaliza el caso de estudio.

Detalle caso general, ejemplo Matlab

[496: dsvd2] Desacoplamiento SVD (2): sintonizado inicial de reguladores y conclusiones

Materiales: [DesacSVD2.pdf]

**** 11:18

[YouTube ▶]

Este vídeo discute un diseño inicial de reguladores desacoplamiento SVD (los dos primeros minutos dan un vistazo rápido al vídeo [489]), si el sistema es predominantemente de primer orden a bajas frecuencias: se dan guías de diseño de un regulador PI para pruebas iniciales: si todo está escalado, ganancia “1” del regulador parece razonable como primer intento.

Al final, se resume la metodología y se presentan conclusiones.

Nótese, no obstante que la sintonía aquí propuesta es meramente “orientativa” y no es, ni mucho menos, “obligatoria”. En el vídeo [498], por ejemplo, se propone utilizar el código de sintonía automática [pidtune](#) para sintonizar PIs con un objetivo de ancho de banda dado por el cruce con 0dB del **sigma** de la maniobra asociada.

[497: dsvdml] Desacoplamiento SVD: ejemplo Matlab

Materiales: [[CÓD.:](#) desacopsvd.mlx] [[PDF](#)]

**** 15:45

[YouTube ▶]

Este proceso presenta un código Matlab donde se detalla la metodología de los vídeos [489] y [496], sobre una planta de 3 salidas y 4 entradas, en la que el SVD de la ganancia recomienda controlar únicamente dos componentes principales de las salidas pese a tener cuatro actuadores!. Obviamente, si la planta no estuviera construida debería, posiblemente, rediseñarse (resulta muy “chocante” que con cuatro actuadores sólo se pueda controlar dos direcciones –combinaciones lineales– independientes de salidas), pero el desacoplamiento SVD intenta aprovechar “lo que hay” y hacerlo mediante un par de reguladores de tipo Proporcional+Integral.


Por simplicidad, se supone que la planta usada en el ejemplo ya ha sido escalada a ± 1 . El código calcula los SVDs requeridos, los reguladores, y simula la respuesta temporal en bucle cerrado de salidas y de acción de control. También comprueba, en su parte final, que si se hubiera decidido controlar tres salidas entonces los actuadores alcanzarían un valor máximo de alrededor de 10 para corregir el error, que no es admisible.

Otro ejemplo Matlab de desacoplamiento SVD de un prototipo de horno se discute en el vídeo [498], y la implementación Simulink se discute en los vídeos [499] y [500]; el modelado del horno en concreto se discute en detalle en el vídeo [441].

Caso de estudio: horno lineal

[498: forn3] Desacoplamiento SVD de un prototipo de horno lineal: diseño de reguladores y análisis tiempo/frecuencia en componentes principales

Materiales: [[CÓD.:](#) desacsvd3actHORNOa.mlx] [[PDF](#)]

****  10:11

[YouTube ▶]

Este vídeo desarrolla el diseño del control por desacoplamiento SVD de un modelo de horno lineal. El detalle sobre el modelado y el análisis de controlabilidad entrada/salida puede visualizarse en el vídeo [441], aunque aquí se dedica hasta el minuto [02:50] a revisarlo de forma rápida.

La metodología de desacoplamiento SVD se encuentra discutida en teoría en los vídeos [489] y [496]. Otro ejemplo Matlab de un proceso $G(s)$ arbitrario se aborda en el vídeo [497].


Aquí, siguiendo consideraciones parecidas, se sintoniza el regulador con **pidtune** buscando un ancho de banda objetivo igual al determinado en el diagrama de valores singulares **sigma** del modelo (escalado).

Se simula (en las coordenadas dadas por los componentes principales) la respuesta ante escalón de referencia en cada maniobra, y se examina el diagrama de respuesta en frecuencia de error y acción de control (proyectado sobre las direcciones principales de entrada y salida). En la parte final del vídeo se sube un poco el ancho de banda del **pidtune**, hasta que sature la respuesta escalón. Debido a la diferente geometría esferas/cuadrados, faltaría realizar una simulación temporal ante distintas referencias o perturbaciones, con/sin saturación, para comprobar si el diseño tiene prestaciones adecuadas. Por brevedad, ello no se ha realizado aquí.

El diagrama de bloques en Simulink de este diseño se describe en el vídeo [499], y se simula en el [500].

[499: forn4] Desacoplamiento SVD de un horno lineal: diagrama Simulink

Materiales: [[CÓD.:](#) dsdvdform.slx] [[PDF](#)]

***  09:32

[YouTube ▶]

Este vídeo presenta el diagrama de bloques Simulink en el que se implementa el control PI con desacoplamiento SVD del vídeo [498], de modo que los dos reguladores PI aparezcan *explícitamente*, para poder resintonizarlos o añadir antiwindup. El diagrama es equivalente, con variaciones mínimas, a los descritos en la revisión teórica del vídeo [489]. La simulación propiamente dicha de este diseño aparece en el vídeo [500].

[500: forn5] Desacoplamiento SVD de horno lineal: simulación en Simulink, análisis de respuesta

Materiales: [[Cód.: dsvdfor.slx](#)] [[PDF](#)]

**** 11:56

[YouTube ▶]

Este vídeo presenta los resultados (respuesta temporal) de la simulación de un modelo de horno con desacoplamiento SVD. Los pasos previos de modelado y diseño de control se detallan en los vídeos [441] y [498]; el significado de los diferentes elementos del diagrama de bloques se detalla en el vídeo [499]. Aunque se resume todo rápidamente en el primer minuto, la visualización previa de dichos contenidos es recomendada.

Primero, se introducen referencias en exactamente las direcciones de salida asociadas al SVD de la ganancia, comprobando que son seguidas sin error, y comprobando el grado de saturación según las diferentes amplitudes probadas.

La parte final del vídeo introduce referencias “físicas” y se comprueba que lo que se sigue es la proyección sobre el subespacio 2D que se ha elegido controlar. Se comprueba el significado de dicha proyección y que cuando la referencia es “rara” no se sigue, como intencionalmente se pretendía al diseñar el par de PIDs.

14.4 Desacoplamiento/linealización por realimentación del estado

Nota: Existe una metodología de “desacoplamiento por realimentación del estado”, que por brevedad *no* está cubierta en el vídeo [ectdc(19:49)]. Es un material clásico en textos de teoría de ingeniería de control, a los que se remite al lector, tanto en el caso lineal como en el no-lineal (en el que da lugar a una herramienta estándar “par calculado” en robótica). De hecho el caso lineal es un *corolario* de una técnica básica de control no lineal que se denomina *linealización por realimentación*.

Aunque no se disponga de materiales extensos sobre estas técnicas, no obstante, las ideas básicas de desacoplamiento y linealización por realimentación del estado pueden ser comprendidas de forma preliminar e intuitiva visualizando los materiales a continuación.

Ejemplo 1: proceso de orden 2 Aplicaremos la técnica a un primer ejemplo de forma manual, sin Matlab.

[501: fblin1] Linealización y desacoplamiento por realimentación: ejemplo 2
entradas 2 salidas (sin Matlab)

Materiales: [LinRealEj1Esp.pdf]

**** 13:45

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute un ejercicio académico sencillo de calcular las ecuaciones del desacoplamiento y linealización por realimentación de un sistema de 2o orden, con 2 entradas y 2 salidas, grado relativo 1 en ambas (las entradas aparecen con primeras derivadas de salidas). El sistema tampoco tiene singularidades ni “dinámica cero” a analizar.

La *linealización por realimentación* es la generalización de lo que en casos muy sencillos llamaríamos linealizar haciendo un *cambio de variable*, p.ej., si sale un cuadrado de la entrada, hacer una raíz cuadrada a la salida de un controlador... la idea se ilustra en un péndulo giratorio en el vídeo [93], por ejemplo.

Revisa el vídeo [915] o el ejemplo Matlab [916] si has olvidado las manipulaciones con derivadas parciales y regla de la cadena multivariable.

Otro ejemplo académico de estas ideas en sistemas lineales (persiguiendo, por tanto, únicamente el desacoplamiento) se aborda en el vídeo [503], y ejemplos aplicados a tanque de mezclado y robot móvil se discuten en los vídeos [502] y [505]. Se aconseja visualizar los tres vídeos referidos para comprender mejor la metodología y cómo no puede aplicarse a procesos de fase no mínima.

Ejemplo 2: procesos químicos no lineales Una aplicación de la técnica a control de “procesos” es la que sigue:

[502: mixfl]

Linealización y desacoplamiento por realimentación del estado:
ejemplo tanque de mezclado (Matlab)

Materiales: [[CÓD.:](#) DesacoplRealimentTanqueMezclado.mlx] [[PDF](#)]

***** 08:02

[YouTube ▶]

Este vídeo discute la linealización y desacoplamiento por realimentación del estado en un tanque de mezclado (mediante elementos no lineales, al ser el proceso no lineal). Un ejemplo sencillo sin usar Matlab se aborda en el vídeo [501], cuya visualización previa se aconseja.

El tanque de mezclado ha sido modelado en detalle en el vídeo [22]. No obstante, el primer minuto y medio en el vídeo aquí discutido revisa rápidamente dicho modelo, por comodidad del lector, dado que, obviamente, el modelo no lineal es necesario conocerlo para poder programar la cancelación de sus no-linealidades. Si se quisiera evitar usar expresiones no lineales, el modelo linealizado y el desacoplamiento alrededor de un punto de funcionamiento usando dicho modelo linealizado se abordan, por ejemplo, en los vídeos [95] y [487], respectivamente. Si las referencias de concentración cambiasen mucho, el uso de ecuaciones no lineales sería ventajoso (por eso el *ratio control* se usa con profusión en procesos químicos); si todo fueran pequeños incrementos alrededor de un punto nominal de operación, entonces no habría demasiada diferencia entre el enfoque no-lineal general y el desacoplamiento linealizado.

Tras el modelado, se entra en el objetivo principal del vídeo que es linealización y desacoplamiento. En concreto, primero se calculan las expresiones $\frac{dx_A}{dt} = \phi_1(v_a, v_b, q_{ain}, q_{bin})$ y $\frac{dh}{dt} = \phi_2(v_a, v_b, q_{ain}, q_{bin})$ mediante la Symbolic Toolbox, multiplicando el Jacobiano de la ecuación de salida ([jacobian](#)) por la ecuación de estado.

Después, se intenta resolver el sistema de dos ecuaciones

$$\phi_1(v_a, v_b, q_{ain}, q_{bin}) = \nu_1, \quad \phi_2(v_a, v_b, q_{ain}, q_{bin}) = \nu_2$$

considerando ν_1 y ν_2 como unas nuevas entradas “artificiales” que serán calculadas con posterioridad, y considerando q_{ain} y q_{bin} como incógnitas a “despejar”. Con el comando [solve](#) se consigue, sin ningún tipo de problema, despejar expresiones $q_{ain} = \sigma_1(v_a, v_b, \nu_1, \nu_2)$ y $q_{bin} = \sigma_2(v_a, v_b, \nu_1, \nu_2)$. Usando esas expresiones como ley de control no lineal, se consigue $\frac{dx_A}{dt} = \nu_1$, $\frac{dh}{dt} = \nu_2$ transformando el tanque de mezclado en dos sistemas lineales, desacoplados, de primer orden cada uno de ellos. Lo único que quedaría para completar el diseño serían, por ejemplo, un par de reguladores proporcionales para el proceso $\frac{1}{s}$ (integrador puro), no discutidos en este vídeo.

Otro ejemplo de este tipo de manipulaciones para un problema de control cinemático de robots aparece en el vídeo [505], y la misma idea aplicada a procesos lineales (para desacoplar) se discute en el vídeo [503].

Nótese que en un caso general, el número de derivadas de las salidas a hacer hasta que se pueda plantear el sistema de ecuaciones a resolver con [solve](#) puede ser diferente de uno (aunque en este caso ha sido así para ilustrar un ejemplo sencillo), e incluso puede ser diferente en unas salidas u otras, como ilustra el vídeo [503] arriba mencionado; eso puede ocurrir tanto en un caso lineal como en un caso no lineal. La metodología acaba transformando un proceso de m entradas y m salidas a [blkdiag](#)($1/s^{\rho_1}, 1/s^{\rho_2}, \dots, 1/s^{\rho_m}$), siendo ρ_i lo que se denomina *grado relativo* de la salida i ; en este caso sencillo ha sido $\rho_1 = 1$, $\rho_2 = 1$.

Ejemplo 3: corolario sistemas lineales La particularización al caso lineal (sólo desacoplamiento) se discute a continuación basada en un ejemplo:

[503: fbdclin]

Desacoplamiento por realimentación en sistemas lineales
(ejemplo Matlab)

Materiales: [[CÓD.:](#) PruebaZeroDynamicsLineal.mlx] [[PDF](#)]

***** 14:42

[YouTube ▶]

Este vídeo presenta un par de casos de estudios donde se aplica la misma metodología que el desacoplamiento+linealización por realimentación (general, incluso para sistemas no lineales), pero dado que el sistema al que se le aplica es ya lineal, el resultado únicamente es desacoplamiento. Un ejemplo “manual” de la metodología sin utilizar Matlab se aborda en el vídeo [501], sobre un proceso no lineal, que podría ser conveniente visualizar previamente.

Es importante darse cuenta que la técnica implícitamente lleva asociada una *cancelación* de la dinámica y que NO puede aplicarse a procesos de fase no mínima, como se discute en la parte final de este vídeo.

Además, para ilustrar las posibles dificultades, se usa un modelo de sistema donde cada una de las salidas tiene diferente grado relativo, con lo que el número de derivadas a hacer en cada una de ellas es diferente. La metodología acaba transformando un proceso de m entradas y m salidas a $\text{blkdiag}(1/s^{\rho_1}, 1/s^{\rho_2}, \dots, 1/s^{\rho_m})$, siendo ρ_i lo que se denomina *grado relativo* de la salida i ; en este ejemplo, resulta $\rho_1 = 2, \rho_2 = 1$.

La metodología es básicamente idéntica a las de los vídeos [502] o [505]; lo que ocurre es que en el caso lineal no es necesario usar “matrices jacobianas de derivadas parciales” y la Symbolic Toolbox, dado que todo puede calcularse equivalentemente mediante las matrices (A,B,C,D) de la representación interna. En muchos libros, los jacobianos se utilizan en los temas de control no lineal y las operaciones con (A,B,C,D) en los temas de control lineal, pero la idea subyacente es exactamente la misma, y el caso lineal es simplemente un caso particular del caso general no-lineal.

La parte final del vídeo aplica la metodología a un proceso de fase NO mínima. En ese caso, queda una dinámica no observable que es inestable (el polo que cancela el cero para que no aparezca en la Matriz de Transferencia final con sólo integradores) y, por tanto, el resultado serían sistemas que NO son internamente estables y que no podrían funcionar en la práctica. Para comprender mejor el significado físico del concepto de dinámica interna no observable se recomienda visualizar el vídeo [507] donde se analiza dicha dinámica en un problema de robótica móvil.

14.4.1 Robótica móvil: control por punto descentrado, caso de estudio

[504: robdmod]

Modelado cinemático de un robot diferencial de 2 ruedas no holonómico para control por punto descentrado

07:02

Materiales:

[CÓD.: ControlPuntoSeparadoRobotMovil.mlx]
[PDF]

[YouTube ►]

Este vídeo modela un robot de dos ruedas tipo diferencial, obteniendo ecuaciones de estado y de salida (no lineales). Las variables de entrada son las velocidades angulares de cada rueda, y las salidas son las dos coordenadas cartesianas (movimiento plano) de la posición de un punto descentrado separado del eje que une las ruedas.

En efecto, el robot es no holonómico, por lo que no se pueden hacer movimientos de puntos del eje en la dirección del mismo. Este tipo de restricción es discutido informalmente en el vídeo, comentando que para superarlas se necesitan maniobras tipo “aparcamiento” que se modelan como el *Lie bracket* en control no lineal, fuera de los objetivos de este material.

[505: robdf]

Desacoplamiento y linealización por realimentación de un robot de 2 ruedas para control cinemático

10:11

Materiales:

[CÓD.: ControlPuntoSeparadoRobotMovil.mlx]
[PDF]

[YouTube ►]

Este vídeo plantea el desacoplamiento y linealización por realimentación de un robot no holonómico de tipo diferencial cuyo modelado se revisa en los primeros dos minutos de este

vídeo, pero se detalla en el vídeo [504]. UN ejemplo académico sencillo de la metodología, sin usar Matlab, se aborda en el vídeo [501], cuya visualización previa se aconseja.

Basándose en la regla de la cadena, se calculan las derivadas de las salidas (posición de un punto descentrado) mediante la multiplicación del Jacobiano de la ecuación de salida por la ecuación de estado. Ello obtiene la cinemática directa de velocidades de rueda a velocidades de punto que desea ser controlado. El comando `solve` invierte dicha expresión y calcula las velocidades de rueda que obtendrían (para un ángulo conocido de orientación) unas velocidades lineales prefijadas del punto de control.

Con todo ello, el proceso se expresa como dos integradores (el control cinemático de robots resulta en grado relativo 1). Otro ejemplo donde las mismas ideas se aplican a un modelo de un tanque de mezclado se desarrolla en el vídeo [502]. El control “dinámico” de robots (calcular las fuerzas y pares para conseguir aceleraciones prefijadas) generaliza esta idea y resulta en modelos de grado relativo 2; en ese caso la metodología se denomina *control por par calculado* (computed torque, en los textos en lengua inglesa).

La simulación de un control cinemático en el que el desacoplador aquí desarrollado forma parte se detalla en el vídeo [506], cuya visualización se aconseja.

El vídeo [507] demuestra que la estrategia aquí presentada no puede ser utilizada para maniobrar el robot “marcha atrás”.

[506: robd2gls] **Control cinemático por punto descentrado de robot de dos ruedas (proporcional + 2GL con feedforward de velocidad de trayectoria), simulación**

Materiales: [[CÓD.: robd2gls.zip](#)] [[PDF](#)]

**** 17:00

[YouTube ▶]

Este vídeo simula la linealización y desacoplamiento por realimentación del estado para un robot de dos ruedas desarrollado en el vídeo [505], donde mediante cierto cambio de variable y desacoplador se transforma el sistema a un par de integradores simples desacoplados `blkdiag(1/s, 1/s)`.

En concreto, se comprueba que un control externo de esos supuestos integradores de tipo proporcional consigue error cero en el posicionamiento ante referencias constantes (como era de esperar). Sin embargo, no puede seguir sin error trayectorias rectilíneas debido al “error de velocidad” de los bucles cerrados “tipo 1” (con un único integrador). En trayectorias de geometría más compleja, ese error de seguimiento se hace todavía más patente.

Para solucionarlo, se propone una estructura de 2 grados de libertad donde además del componente proporcional al error, se añade un componente igual a la velocidad de la trayectoria que debe ser seguida. El vídeo discute las ventajas de esta opción y se comprueba en simulación que entonces ya pueden ser seguidas sin error trayectorias de geometría compleja (suponiendo, claro está, que no hay error de modelado, saturación, deslizamiento, etc.).

Nota: en todas las simulaciones, si las condiciones iniciales hacen que se comience “marcha atrás”, de todos modos el robot se “da la vuelta” y se pone hacia “adelante” (el punto descentrado pasa por un punto de la trayectoria “antes” que el eje de las ruedas). Esto es previsible dado que puede demostrarse que la marcha atrás es inestable con la estrategia de desacoplamiento/linealización aquí simulada, según detalla el vídeo [507].

El controlador realimentado en este caso es estático (proporcional), al ser la linealización un sistema de primer orden para cada grado de libertad. Si el controlador fuera dinámico (por querer controlar aceleraciones o/y filtrar algún ruido) entonces el código debería ser modificado, adaptando a este entorno código con ecuaciones de estado y salida de controlador, del estilo de lo discutido, por ejemplo, en el vídeo [355].

[507: robdzd] **Dinámica no observable en control punto descentrado robot diferencial de 2 ruedas no holonómico**

Materiales: [[CÓD.: ControlPuntoSeparadoRobotMovil.mlx](#)] [[PDF](#)]

***** 14:29

[YouTube ▶]

Un robot no holonómico de 2 ruedas con control por punto descentrado puede transformarse a dos integradores, ver vídeos [505] (teoría) y simulaciones en [506].


No obstante, dado que el orden del modelo original es tres, existe un estado “perdido” que se ha convertido en una dinámica no observable. Este vídeo analiza dicha dinámica y comprueba que en trayectorias rectilíneas el robot se alinea con la recta seguida por el punto de control, y existen dos puntos de equilibrio del ángulo (marcha atrás, marcha adelante) pero sólo es estable la dirección “hacia adelante”.

Esta dinámica interna no observable resulta más sencilla de analizar, obviamente, si estas metodologías se aplican a un sistema lineal que debe ser estabilizado en el origen, como ocurre en el caso del vídeo [503], dado que entonces sólo hay un único punto de equilibrio que será estable o inestable, en vez de los dos equilibrios que aquí aparecen; las ideas generales, no obstante, son las mismas.

14.5 Control en cascada

[508: ectcas] Estructuras de control multivariable: control en cascada
sensor/actuador extra

Materiales: [CtrlDescen.pdf]

**  19:03

[YouTube ▶]

- Control en cascada con sensor extra: diagrama de bloques. Ejemplo servoválvula. Ventajas.


Nota: En los vídeos [656] y [657] se aborda la sintonía de estos controladores en cascada mediante control óptimo. Con separación de escala de tiempos, no obstante, en la gran mayoría de las aplicaciones prácticas se sintonizan PID maestro/esclavo de forma manual. Los vídeos referidos están indicados para estudiantes avanzados/especialidad, sin pretender que sean la opción más recomendable en la práctica en un caso general, que casi siempre es la sintonía manual separada: no te los tomes muy en serio, que el control en cascada es super-fácil.

- Control con actuador extra: diagrama de bloques. Ejemplo intercambiador con by-pass. Ventajas. El vídeo [509] presenta un ejemplo Matlab con PID.

Un análisis complementario del control en cascada actuador extra y su relación con el control gradual se discute en el vídeo [511], que se recomienda visualizar.

[509: dactcasc] Control PID en cascada con actuador extra: ejemplo Matlab

Materiales: [Cód.: cascadaSobraActPID.mlx] [PDF]

***  09:59

[YouTube ▶]


En este vídeo se simula una configuración de control en cascada con actuador extra utilizando dos PID (sintonizados por prueba y error) sobre el proceso analizado en el vídeo [445], para ilustrar las capacidades de la estructura comparada con el uso de un único actuador.

Nota: Un ejemplo Matlab de la sintonización con metodología de control óptimo \mathcal{H}_∞ de este mismo proceso de dos actuadores es abordada en los vídeos [658] y [659]; obviamente, resulta de interés comparar las respuestas conseguidas.

14.6 Estructuras mixtas y otras propuestas

[510: dosact1] Control con exceso de actuadores (1): equilibrado de carga,
rango dividido, override

Materiales: [TwoActuatorsResumen.pdf]

***  19:13

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

En este vídeo se discuten estrategias para estructuras de control con 1 variable controlada y 2 manipuladas. Los tres primeros minutos motivan el problema introduciendo ejemplos prácticos de donde aplica (intercambiador con bypass, neutralización, calentamiento por etapas, ...), y plantea que hay dos grandes divisiones de las estrategias para solventar el problema: aquellas con un sólo controlador en funcionamiento y aquellas donde se genera una referencia y una variable controlada secundarias, y existen, por tanto, dos controladores activos.

En este primer vídeo, se discute la primera de las opciones (un sólo controlador activo) y se explican las estructuras de carga equilibrada (load balancing), rango compartido (split range) y override (dos controladores, pero uno sólo activo en emergencias; realmente override está un poco ‘en medio’ y dependiendo de opiniones podríamos decir que hay un bucle o dos bucles).

Un segundo vídeo [511], continuación de éste, explica las estrategias con referencia secundaria (cascada, gradual, ...).

[511: dosact2]

Control con exceso de actuadores (2): cascada actuador extra, gradual, doble cascada

Materiales: [TwoActuatorsResumen.pdf]

**** 17:54

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

En este vídeo se continua la discusión sobre estrategias para estructuras de control con 1 variable controlada y 2 manipuladas.

Hay dos grandes divisiones de las estrategias para solventar el problema: aquellas con un sólo controlador en funcionamiento y aquellas donde se genera una referencia secundaria y existen dos controladores activos.

Los tres primeros minutos dl vídeo revisan las estrategias con un controlador activo: carga equilibrada (load balancing), rango compartido (split range) y override ; estas estrategias se discutieron en detalle en el vídeo [510], previo a éste.

Continuación de dicha exposición, aquí se explican las estrategias con referencia secundaria. Se proponen:

- control en *cascada con actuador extra*, donde la referencia secundaria es un punto de operación de una de las variables manipuladas.
- control *gradual* donde, si se dispone de un sensor intermedio, se puede plantear un calentamiento, neutralización, etc. por *etapas*, con una referencia secundaria para ese sensor intermedio que una especie de multiloop debería mantener.
- Control *doble cascada* donde si se dispone de actuador extra y también de sensor extra, entonces la referencia para el sensor intermedio ‘gradual’ es generada por la estructura ‘cascada actuador extra’ maestra.

Estas estructuras en cascada funcionan mejor (facilidad de sintonía) cuando hay una “separación de escalas de tiempo”, mientras que las estructuras del primer vídeo funcionan mejor cuando ambas variables manipuladas tienen anchos de banda comparables y dinámicas similares.

[512: ectmix]

Estructuras de control multivariable: estructuras mixtas (ejemplo intercambiador)

Materiales: [CtrlDescen.pdf]

*** 15:22

[YouTube ▶]

Este video presenta casos de estudio donde se combinan estructuras mixtas con varios niveles jerárquicos en cascada, multibucle o desacoplamiento.

Se basa en el ejemplo de intercambiador usado en el video [508], al que se le añaden sensores y actuadores adicionales para mejorar prestaciones y se presenta una estructura

con múltiples reguladores tipo PID, prealimentación o desacoplamiento que aprovecha esos sensores y actuadores. Otras estructuras mixtas se discuten, por ejemplo, en los casos de estudio de los videos [531], [529], [530]: en efecto, es importante darse cuenta de que la mayoría de aplicaciones de control industrial avanzado se construyen combinando distintas estructuras básicas.

[513: ratc]
Ratio control (ejemplo neutralización) [English]

*** 10:30

[PLAY ▶]

Este vídeo introduce el concepto de *ratio control* o *control de razón* en un caso de neutralización de un ácido (control de PH).

Nota: Este vídeo está en lengua **inglesa**.

[514: ectoth]
Estructuras de control multivariable: otras opciones (override, ratio, etc.)

*** 18:17

Materiales: [CtrlDescen.pdf]

[YouTube ▶]

Este video presenta (muy brevemente) estructuras adicionales a las de los videos anteriores, que también pueden ser utilizadas para diseñar estructuras de control en plantas multivariables.

- Control indirecto
- Control con selectores (override)
- Control split-range (calor/frío)
- Control gradual; un detalle de estructura de control gradual en un ejemplo de neutralización puede revisarse en el vídeo [529].
- Tolerancia a fallos: redundancia, votos, override.

Nota: la estructura de control denominada *ratio*-control, importante en muchos procesos químicos y de mezclado, se discute en el video [531] aplicada a control de combustión.

[515: infec]
Control inferencial (ejemplo destilación)
Gregory Reeves (NCSU)

*** 10:29

[PLAY ▶]

En este vídeo se discute el control *inferencial*, y se pone un ejemplo de control inferencial donde una concentración de propano se podría controlar de forma indirecta (inferencial) controlando una temperatura (variable secundaria) que correlaciona bien con la variable primaria a controlar, prescindiendo de un analizador en linea costoso y con un retardo excesivo.

Nota: Este vídeo está en lengua **inglesa**.

[516: GcomPID]
Expresión de controladores genéricos de orden bajo como reguladores PID

*** 10:08

Materiales: [RegulComoPID.pdf]

[YouTube ▶]

En algunos casos, se diseñan reguladores mediante técnicas como modelo interno, cancelación, reguladores algebraicos, lugar de las raíces, óptimo, que dan lugar a un regulador genérico $C(s)$. Si $C(s)$ es de primer o segundo orden (o, siendo de orden mayor, pueden eliminarse dinámicas rápidas y aproximarlos a orden 2), entonces es posible en bastantes casos expresarlo como un PID. Esto tiene ventajas de implementación práctica (en PLCs o PIDs industriales) y facilita la comprensión intuitiva de su funcionamiento. Este vídeo discute algunas opciones en dicho sentido. Como ejemplo de estas ideas, el vídeo [403] utiliza la idea del control por modelo interno (vídeo [391]) para diseñar reguladores PIDs para plantas de bajo orden.

14.7 Control con dos grados de libertad

[517: c2gl1] Control con dos grados de libertad: planteamiento del problema y motivación

***  08:19

Materiales: [Control2GL.pdf]

[YouTube ►]

Este video presenta el control con dos grados de libertad (2GL) $u = K_r(s) \cdot r + K_y(s) \cdot y$ (procesado independiente de referencia y salida), más general que la estructura de “un grado de libertad” con realimentación del error $u = K_e(s) \cdot (r - y)$ (control 1GL). Justifica su uso en términos de conseguir un buen compromiso entre diversos factores que simultáneamente deben ser solventados en un buen diseño de control:

- robustez (una K_y o K_e baja en procesos estables garantiza estabilidad: si G es estable y no toco nada, no rompo nada),
- seguimiento de referencias (K_e alta en 1GL para error pequeño), pero tenemos la alternativa de control en bucle abierto ($K_y = 0$, $K_r = G^{-1}$), o cosas “intermedias” que es precisamente el 2GL.
- rechazo de perturbaciones (K_r no le afecta, sólo K_y o K_e intervienen en esas prestaciones).

Como ejemplo de estos compromisos de diseño, se recomienda al lector visualizar el vídeo [589] para un ejemplo de control óptimo (Simulink Response Optimization) donde los parámetros óptimos ante cambio de referencia tienen un comportamiento no satisfactorio ante perturbaciones en un cierto rango de frecuencias, y viceversa.

[518: c2gl1siso] Control con dos grados de libertad en reguladores con estructura sencilla (SISO, PID, ...)

***  12:18

Materiales: [Control2GL.pdf]

[YouTube ►]

Según se argumenta en el vídeo [517], el control con dos grados de libertad (2GL) $u = K_r(s) \cdot r + K_y(s) \cdot y$ (procesado independiente de referencia y salida), permite conseguir determinados compromisos en prestaciones ante referencia, perturbaciones y errores de modelado que la estructura de “un grado de libertad” con realimentación del error $u = K_e(s) \cdot (r - y)$ (control 1GL) no puede alcanzar.

Este vídeo discute su implementación, en estructuras convencionales (casi siempre, tipo PID en aplicaciones), como preprocesado de referencia $u = K(s)(Q(s)r - y)$, siendo $K(s)$ un regulador sintonizado para cancelar perturbaciones de forma óptima, y siendo $Q(s)$ un filtro que modifica la forma de onda de la referencia para intentar mejorar el comportamiento ante cambios de r del regulador 1GL (el caso con $Q(s) = 1$). Se plantea $Q(s) = K_e \frac{\alpha s + 1}{\beta s + 1}$, por ejemplo. Realmente en muchas aplicaciones el preprocesado de referencia incluye términos no lineales adicionales a ese tipo de filtrado: muchas veces se necesita saturar a límites “seguros para cada aplicación” tanto la *amplitud* de la referencia como su *tasa de cambio* (rate saturation).

Complementariamente al prefiltrado de referencia $Q(s)$, un elemento de control 2GL *muy utilizado* es, en un PID, implementar acción derivada sólo en la salida: $u = (K_p + \frac{T_i}{s}) \cdot (r - y) + T_d \frac{s}{\tau_F s + 1} y$. De ese modo se evita el “*derivative kick*” (*patada* derivativa) que haría que el actuador sature durante un cierto tiempo tras un cambio en escalón de referencia si la acción derivada fuese $T_d \frac{s}{\tau_F s + 1} \cdot (r - y)$; este incremento brusco es perjudicial en muchas aplicaciones. Un ejemplo de sintonizado de un control PID 2GL donde no se derive la referencia (para evitar la “patada” o “*derivative kick*” ante cambios escalón de la misma) aparece en el vídeo [588].

Este vídeo finaliza con un breve ejemplo numérico: sintonizando un PD para rechazar perturbaciones (optimizando un criterio de *mínima varianza*) y luego preprocesando referencia para conseguir seguimiento de ella mejor, sin sacrificar la *varianza mínima* conseguida ante perturbaciones.

[519: c2glgeneric] Control con dos grados de libertad en reguladores genéricos multivariables
07:03

Materiales: [Control2GL.pdf]

[YouTube ▶]

Este video presenta el control con dos grados de libertad (2GL) con reguladores genéricos (modelos de referencia, prealimentación, inversas de matrices de transferencia).

Primero se propone un caso de control 2GL donde las variables controladas coinciden con las variables medidas. Si se desea un comportamiento ante referencia $y = T_r r$, se propone un regulador $u = u_F + R(s) \cdot (y - y_F)$, donde $u_F = G^{-1} T_r r$ e $y_F = T_r r$. Esto generaliza el caso del punto de funcionamiento constante en una linealización, donde $u_F = u_{eq}$ constante, y el regulador actúa ante incrementos de salida $y - y_{eq}$, siendo y_{eq} el punto de linealización constante. En este caso de control 2GL, u_f e y_F son trayectorias.

Nota: de hecho, la idea de arriba es muy similar a la que motiva la linealización alrededor de trayectorias (vídeos [875], [876]), y la relación entre control 2GL y dicha linealización es discutida específicamente en el vídeo [877].

La parte final del vídeo discute un caso genérico donde las variables controladas son z y las medidas son y , intencionadamente diferentes. En ese caso, si se desea $z = T_r r$, entonces $u_F = G_{zu}^{-1} T_r r$ seguiría la trayectoria deseada para z sin errores de modelado, y se debería realimentar la diferencia entre las medidas y y la simulación de dichas medidas $y_F = G_{yu} u_F = G_{yu} G_{zu}^{-1} T_r r$, construyendo un regulador que generara $u = u_F + R(s) \cdot (y - y_F)$ como en el caso anterior.

También existen estructuras generalizadas de control óptimo que incorporan de forma natural el control con 2 grados de libertad; por ejemplo, tenemos el control predictivo donde se le informa de la referencia futura, o controles \mathcal{H}_∞ con la adecuada planta generalizada según se discute en el vídeo [663]. De todos modos, por brevedad, esos diseños no son objetivo del presente material.

El vídeo finaliza con unas conclusiones generales que resumen las ideas principales de los vídeos [517], [518] y el aquí discutido.

Ejemplos de control 2GL basado en optimización se abordan en el vídeo [663], basados en control por modelo interno en el vídeo [520] y una aplicación robótica en el vídeo [506].

[520: imc2gl] Control por modelo interno de 2 grados de libertad
Materiales: [IMC62gl.pdf]

*** 09:19
 [YouTube ▶]

Este vídeo detalla la extensión del control por modelo interno IMC (vídeo [391]) al caso del control con 2 grados de libertad, cuya motivación práctica se discute en el vídeo [517].

Aunque en estructuras genéricas de control (por ejemplo, con plantas inestables) debería abordarse según las ideas del vídeo [519], en el caso particular de G estable y control diseñado por IMC, un par de sencillas modificaciones al diagrama de bloques básico permite transformarlo en una estructura 2GL. Los diagramas de bloques asociados a estas modificaciones son los que se presentan en este vídeo.

14.8 PID avanzado: antiwindup por modo de seguimiento (tracking mode)

[521: awtr1] Controlador PID con antiwindup "tracking mode": ideas básicas y ecuaciones de la dinámica
Materiales: [AWTrack1.pdf]

*** 11:00
 [YouTube ▶]

Este vídeo presenta las ecuaciones de un PID con tracking mode (con realimentación externa) que incluyen un término $\frac{K_t}{s}(u_{EF} - u)$, siendo u_{EF} una entrada externa, bien la propia acción saturada, bien de señales del sistema controlado.

El vídeo motiva el problema del windup (inestabilidad de $\frac{1}{s}e$ en bucle abierto), presenta las ecuaciones en transformada de Laplace (en forma paralelo / ISA), y analiza la dinámica resultante cuando $u = u_{EF}$, que es la de un PID normal, y cuando $u \neq u_{EF}$ (dinámica en modo de seguimiento).

Las aplicaciones concretas de este tipo de regulador se discuten en el vídeo [522], continuación de éste, y la sintonía de K_t en el vídeo [523].

El antiwindup en reguladores genéricos en representación interna se discute en el vídeo [673], y en particular, en el vídeo [674] se demuestra que el tracking-mode discutido aquí es un caso particular de dicho esquema general.

[522: awtr2]

Controladores PID con antiwindup tracking mode: aplicaciones y diagramas de bloques

Materiales: [AWTrack2.pdf]



08:42

[YouTube ►]

En este vídeo se discuten diferentes escenarios de aplicación del PID con "modo de seguimiento" (tracking mode) cuyas ecuaciones se presentaron en el vídeo [521].

En concreto, plantea diagramas de bloques para:

1. Bumpless transfer (conmutación suave entre conectado y desconectado), lo cual también se aplicaría a control con selectores (override)
2. Back-calculation antiwindup, con $u_{EF} = \text{sat}(u)$ implementada directamente en el controlador.
3. Antiwindup con medida física de actuador (no sería necesario conocer los límites)
4. Antiwindup en el bucle maestro de un control en cascada
5. Antiwindup en esquemas con prealimentación y desacoplamiento

Las recomendaciones para seleccionar el valor de K_t o su inversa T_t en cada uno de los casos se discuten en el vídeo [523].

La generalización de este esquema de antiwindup a reguladores genéricos en representación interna (quizás multivariables) se discute en el vídeo [673], y en el vídeo [674] se presenta el tracking-mode en PIDs aquí discutido como un caso particular de dicha metodología genérica.

[523: awtr3]

Controladores PID con antiwindup tracking mode: sintonizado de la dinámica de tracking (T_t)

Materiales: [AWTrack3.pdf]



12:15

[YouTube ►]

Este vídeo, continuación de los vídeos [521] y [522], discute orientaciones cualitativas sobre la selección del parámetro T_t (constante de tiempo en modo tracking).

Primero se discuten las ventajas e inconvenientes de que T_t sea demasiado pequeño o demasiado grande. Una vez analizadas, se dan recomendaciones para antiwindup, para antiwindup en esquemas en cascada, para reguladores con acción derivada, para transferencia suave y para control override.

Muchas referencias de literatura proponen, como valor sugerido para probar inicialmente, $T_t \approx T_i$ o $T_t \approx \sqrt{T_i \tau_d}$. En general, las recomendaciones son relativamente vagas y su comportamiento debería ser confirmado en simulación, sobre todo en los escenarios de secuenciación de operaciones y override.

Como ejemplo de aplicación, el uso de este tipo de antiwindup en un modelo de tanque de mezclado se aborda en el vídeo [551].

Nota: Si la interacción/conmutación entre bucles y sus antiwindups se complica en control *override*, quizás fuese razonable considerar la opción de control *predictivo* (centralizado) que maneja de forma natural las restricciones en entradas y salidas. Sin embargo, si las desviaciones anómalas de salidas se deben a fallos no modelados, el control predictivo podría no proporcionar una solución correcta mientras que un *override* “de emergencia” sí.

14.9 Casos de estudio y ejemplos adicionales

Ejercicio académico de multibucle, desacoplamiento y feedforward

[524: ejrgadc1] **Ejercicio:** RGA/Niederlinski/Desacoplamiento en proceso 3x3 ***  09:18
Materiales: [[Cód.:](#) ejercicioRGANI1.mlx] [[PDF](#)] [YouTube ▶]


Este vídeo presenta una discusión sobre el emparejamiento a elegir en un proceso de 3 entradas y 3 salidas, dada la matriz de ganancia estática. Se usa la metodología de ganancia relativa (RGA) y el criterio de Niederlinski para confirmar la idoneidad del emparejamiento propuesto.

En la segunda parte del vídeo se discute el desacoplamiento en forma inversa, y el diagrama de bloques de su implementación. Dependiendo de la plataforma software donde se ejecutaran los controladores, habría que romper el bucle algebraico insertando, por ejemplo, retardos de un período de muestreo.

[525: ejdcff2] **Ejercicio:** prealimentación (feedforward) en proceso 3x3 ****  09:27
Materiales: [[Cód.:](#) ejercicioRGAFFDC2.mlx] [[PDF](#)] [YouTube ▶]

Este vídeo discute opciones de control feedforward con/sin desacoplamiento preexistente para el proceso estudiado en la primera parte del ejercicio, vídeo [524]. Se presentan las dos formas matemáticamente correctas de resolver el problema (teoría en vídeo [477]) y la forma más sencilla de explicar (sin hablar de matrices a interlocutores no expertos), pero incorrecta (vídeo [476]) si el proceso no es diagonal-dominante, que consiste en utilizar la fórmula diagonal sin que exista desacoplamiento.

14.9.1 Unificación de resultados

[526: nlff2gldec] **Enfoque no-lineal unificado de punto de operación, 2GL, prealimentación y desacoplamiento** ****  15:48
Materiales: [FF2GLDecNL.pdf] [YouTube ▶]

Este vídeo describe un enfoque unificado, dado un sistema no lineal al:

- Cálculo del punto de funcionamiento (equilibrio) para linealizar
- Control con dos grados de libertad (vídeo [518] en caso lineal)
- Prealimentación (vídeo [475] en caso lineal)
- Desacoplamiento (vídeos [484] y [486] en caso lineal)
- Prealimentación desacoplada (vídeo [477] en caso lineal)


La idea básica es que las ecuaciones en equilibrio $y = m(u, d)$, junto con $y = r$ pueden resolverse no sólo al modelar, sino en línea, según se obtengan valores de r (referencias) y d (perturbaciones medibles). Las versiones desacopladas vendrían de intentar resolver de forma independiente la ecuación $r_i = m_i(u_i, u_{resto}, d)$. En caso de saturación, habría que considerar optimización con restricciones. Nótese que estamos hablando sólo de ecuaciones en equilibrio y que podría haber bucles algebraicos en el desacoplamiento.

También es importante notar que las propias referencias podrían calcularse en función de las perturbaciones medibles d optimizando algún índice económico si es necesario cambiar el punto de operación cuando cambia d , según lo discutido en el vídeo [424].

14.9.2 Casos de estudio de procesos industriales

[527: ecprestank1] Control multibucle de un reactor presurizado, propuesta estructura de control (1)

10:27

*** 

Materiales: [ecprestank1.pdf]

[YouTube ▶]

En este vídeo se discute un primer borrador de estructura de control multibucle de una vasija de reacción presurizada, con 3 lazos en bucle cerrado de control de nivel, temperatura y presión, y un control en bucle abierto de concentración ajustando la ratio entre caudales de dos reactivos a un valor de referencia. Se usa control en cascada, generando referencias de caudal a bucles esclavos que gobiernan válvulas de control.


Existen ciertos inconvenientes o posibles mejoras, que serán discutidos en el vídeo [528].

Aunque más o menos se ha seguido la norma ISA-5.1-2009, hay incorrecciones; no soy experto en dichos diagramas P&ID normalizados, mis disculpas.

Otro caso de estudio de control multibucle aparece en el vídeo [532].

[528: ecprestank2] Control multibucle de un reactor presurizado (2): refinamientos (prelim., override, ratio, split range, ...)

13:47

**** 

Materiales: [ecprestank2.pdf]

[YouTube ▶]

Este vídeo complementa el diseño básico de control multibucle en un tanque presurizado descrito en el vídeo [527].

En concreto se añaden:

- estructuras de rango dividido (split range) para usar dos válvulas en control de la presión,
- Un sensor de composición del líquido en el interior del tanque para alimentar a un *ratio* control en bucle cerrado.
- Un sensor de caudal de salida que alimenta a bloques feedforward (prealimentación) para anticiparse a los efectos de incrementos de dicho caudal sobre nivel y temperatura
- Un segundo sensor de presión y controlador (umbral) de seguridad para que, en caso de fallo del control primario de presión, éste segundo sistema tome el mando (override) y se encargue de abrir la válvula de descarga a la atmósfera del gas interior.


Por supuesto, dependiendo de los detalles de las prestaciones requeridas en una aplicación concreta, podrían no ser necesarios algunos de los refinamientos aquí propuestos (incluso podría ser que ninguno fuera necesario y que la estructura básica del primer vídeo mencionado ya tuviera un comportamiento satisfactorio).

*No soy experto en dichos diagramas P&ID normalizados con ISA-5.1-2009; mis disculpas por las incorrecciones.

Otro caso de estudio de estructuras adicionales al multibucle en control se discute en el vídeo [534].

[529: ecneutra]

Estructura de control + P&ID de una etapa de neutralización
 X. Blasco, J.M. Herrero, C. Ramos, A. Sala (UPV)

***  10:51
 [PLAY ►]


Este vídeo describe la estructura de control (gradual, en dos etapas) y diagrama P&ID de una etapa de neutralización en ingeniería química.

Nota: el objetivo es la comprensión del significado de los diagramas y tener un mínimo de semejanza a las ideas que motivan el control en aplicaciones de este tipo, pero no representa un esquema realmente funcional en la práctica, dado que está incompleto y existen otras alternativas de diseño que deberían compararse.

Este vídeo utiliza los materiales docentes de la asignatura ‘Instrumentación y Control Industrial’ de la ETSII (UPV), y está concebido para los alumnos de dicha asignatura, pero se incluye en esta colección como ejemplo adicional.

[530: ecdestil]

Estructura básica de control + P&ID de una columna de destilación
 X. Blasco, J.M. Herrero, C. Ramos, A. Sala (UPV)


***  12:51
 [PLAY ►]

Este vídeo describe la estructura de control y diagrama P&ID de una etapa de destilación. El objetivo es la comprensión del significado de los diagramas y tener un mínimo de semejanza a las ideas que motivan el control en aplicaciones de destilación, pero no representa un esquema realmente funcional en la práctica, dado que está incompleto y existen otras alternativas de diseño no analizadas aquí.

Este vídeo utiliza los materiales docentes de la asignatura ‘Instrumentación y Control Industrial’ de la ETSII (UPV), y está concebido para los alumnos de dicha asignatura, pero se incluye en esta colección como ejemplo adicional.

[531: ectcald]


Estructuras de control multivariable: caso de estudio, caldera de vapor
Materiales: [CtrlDescen.pdf]

***  19:34
 [YouTube ►]

Este vídeo describe los problemas principales en el control de una caldera de vapor para combustión eficiente (control de nivel, presión, entalpía, inquemados en gases de combustión), utilizando múltiples sensores y actuadores en estructuras que combinan elementos multibucle, desacoplamiento, prealimentación, *ratio* control y cascada.

*No soy experto en P&ID normalizado (ANSI/ISA-5.1-2009, etc.) y por tanto dicha norma no se ha aplicado en los esquemas que aquí se presentan, cuyo objetivo es entender el “control”, y no la competencia de elaborar planos técnicos normalizados de forma correcta.

Proceso calentamiento y mezclado (2 tanques)

[532: estrx2a]	Sistema dos tanques mezclado+calentamiento (1): control multilazo (multiloop)	***  11:59
Materiales: [ttceESP.pdf]		[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo plantea el problema de control en un caso de estudio de control de un sistema con dos tanques y dos elementos calefactores, construido para diluir un cierto reactivo en agua y ser suministrado a una cierta temperatura aguas abajo. El caudal demandado será considerado una perturbación. El punto de operación será diluir 1 litro de reactivo en 5 de agua (concentración 1/6 de salida).

Primero se discute la selección de referencias primarias: temperatura y concentración en tanque 2, porque para eso se construye el proceso; también se pone referencia primaria de nivel, por seguridad y para que el tiempo de residencia de los fluidos tenga una menor variabilidad (aunque ese tiempo dependerá del caudal que atraviesa los tanques, por lo que la variabilidad será inevitable).


Segundo, se preseta la existencia de cuatro variables manipuladas: válvula de reactivo, de disolvente, intercambiador en tubería de entrada, calefactor en tanque de salida.

Existen siete sensores, pero en este vídeo primero se plantea elegir como sensores únicamente los de las referencias primarias (temperatura, concentración, nivel) y pensar en una estrategia multilazo. Por “sentido común” se discute que el caudal de agua se use para control de nivel, el caudal de reactivo se usará para concentración y el calentador en tanque de salida se usará para temperatura.

Si se desea un poco más de detalle basado en “modelos” en vez de “sentido común” sobre los emparejamientos para multiloop en el subsistema de nivel y mezclado, se remite al vídeo [533] donde se detalla modelado, linealización y matriz de ganancia relativa para este caso de estudio. Pero, vaya, lo mejor es que el diseño RGA y el sentido común coincidan, obviamente.


El exceso de variables manipuladas y de sensores podrá ser usado para mejorar prestaciones en estructuras cascada, ratio y feedforward que se discuten en el vídeo [534], continuación de éste.

Otro problema de mezclado, se plantea en un caso de estudio que comienza en el vídeo [542], cuya visualización se aconseja.

[533: mzrga]	mezclado + control de caudal (nivel): control multilazo mediante matriz de ganancia relativa (RGA), teoría	***  10:59
Materiales: [ttceESPDecRtJunto2.pdf]		[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute el detalle sobre el posible emparejamiento en un mezclado “estático” para control descentralizado de un proceso con variables controladas de caudal total y de concentración de mezcla mediante el modelado, linealización y cálculo de matriz de ganancia relativa (RGA) de dicho proceso. Se motiva, por ejemplo, en el caso de estudio del vídeo [532], donde se sacan las conclusiones por “sentido común” en vez de basadas en modelos, que es lo que hacemos aquí.

[534: estrx2b]	Sistema dos tanques mezclado+calentamiento (2): control avanzado cascada, feedforward, ratio	****  20:59
Materiales: [ttceESP.pdf]		[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo es continuación del caso de estudio discutido en el vídeo [532] de control de nivel, temperatura y concentración en un sistema de 2 tanques de líquido. En dicho vídeo se planteó el problema y se revisó la propuesta multilazo (justificada con cálculos en el vídeo [533]) como la opción más básica que, si funciona, no requeriría preocuparse de nada más.

Suponiendo que no funciona bien y que se invierte en instrumentación adicional, este vídeo propone utilizar:

- Cascada sensor extra para control de caudales de entrada (y también en bucle de temperatura),
- Feedforward con sensor de caudal de salida,
- Cascada actuador extra para aprovechar dos elementos calefactores en control de temperatura,
- Ratio control para el bucle de concentración
- Desacoplamiento para disminuir el efecto de control de nivel sobre temperatura.

Con ello, se diseña una estructura final que usa 7 sensores y 4 actuadores para el control de 3 referencias primarias.

Para otro caso de estudio similar, se recomienda visualizar el vídeo [550].

Los vídeos [535] y [536] abordan una discusión opcional sobre las relaciones entre desacoplamiento en forma inversa, ratio control, y desacoplamiento en forma directa. De todos modos, la discusión es, como se ha dicho, meramente opcional, especializada, y que no es necesaria para entender lo que aquí se propone.

[535: mzratdc1]	mezclado estático+control caudal: ratio control, relación con multilazo y desacoplamiento	*****	⌵	11:44
Materiales: [ttceESPDecRtJunto2.pdf]				[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo continua el análisis basado en modelos teóricos del mezclado estático a la entrada de un proceso de 2 tanques comenzado en el vídeo [533] donde se modeló, linealizó y obtuvo el emparejamiento recomendado para control multilazo basándose en la matriz de ganancia relativa (RGA).

Aquí se discute el *ratio* control en concentración, comenzando con justificación meramente intuitiva (similar a la argumentación hecha en el vídeo [534], sin ecuaciones). Pero el objetivo principal de este vídeo es el análisis teórico de dicho ratio control, en concreto su linealización: se observa que la linealización del ratio control puede ser entendida como un multilazo + desacoplamiento (forma inversa) del efecto del bucle de nivel sobre el de concentración. Con esa interpretación, se justifica que el control multilazo de concentración se obtendría del control ratio multiplicándolo por el caudal nominal de agua.

La relación con el desacoplamiento es discutida en más profundidad en el vídeo [536], continuación de éste, que se recomienda visualizar.

[536: mzratdc2]	mezclado estático+control caudal: desacoplamiento inverso y directo, lineal y no lineal	*****	⌵	19:56
Materiales: [ttceESPDecRtJunto2.pdf]				[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo complementa el vídeo [535] en el que se propuso una estructura *ratio control* y se comprobó que su linealización podía ser interpretada como un desacoplamiento (parcial) en forma inversa.

Aquí, inspirados en ello, se propone el desacoplamiento en forma inversa completo (aparece, claro, un bucle algebraico o parásito) en forma lineal y se comprueba que también puede modificarse la estructura ratio del vídeo anterior para incorporar ese desacoplamiento en forma inversa completo *NO lineal*.

La última parte del vídeo discute el desacoplamiento en forma “directa” (sin bucles hacia atrás) mediante inversa de matrices en el caso lineal, o mediante unos cálculos muy fáciles con proporciones en el caso no lineal.


El desacoplamiento inverso crea un bucle parásito que hay que romper/comprobar estabilidad pero por contra protege en cierto modo contra saturación (ver antiwindup en mezclador en vídeo [487]); el desacoplamiento directo no tiene bucles parásitos, pero en caso de saturación las proporciones podrían cambiar.

La generación de la “inversión” de las ecuaciones de mezclado a un caso “dinámico” se llama linealización por realimentación, y un ejemplo de la misma también en un proceso de mezclado (dinámico) se discute en el vídeo [502].

Para otro caso de estudio similar, se recomienda visualizar el vídeo [550].

14.9.3 Tanque de mezclado (V2) con concentración no homogénea y retardo


Modelado

[537: tmrmod1] [Modelo procesado de mezclado en un tanque \(1\): división en dos volúmenes con concentración diferente](#) ****  09:42
Materiales: [[CÓD.:](#) ModeladoTanqueMezclado2.mlx] [[PDF](#)] [\[YouTube ▶\]](#)

En este vídeo se modela un tanque de mezclado, similar al del vídeo [22], pero incorporando una dinámica adicional en el mezclado (concentración interior del tanque no homogénea).

Este vídeo discute aspectos preliminares; en concreto, los detalles de cómo modelar la concentración no homogénea mediante la división del volumen interno en dos volúmenes (cada uno la mitad del total) y cómo calcular los caudales que deben intercambiarse entre suministros de entrada, salida y entre los dos sub-volúmenes. El modelo físicamente perfecto dividiría en muchos más *elementos finitos* e incorporaría ecuaciones de mecánica de fluidos, pero dicho modelado está fuera de los objetivos del presente material.

En este vídeo sólo se considera el volumen “total” y no el volumen de componente “A” que influirá en las concentraciones; el modelo completo incluyendo concentraciones se aborda en el vídeo [538], continuación de éste.

[538: tmrmod2] [Modelo tanque de mezclado con dinámica de mezclado \(2\): modelo en representación interna](#) ****  10:02
Materiales: [[CÓD.:](#) ModeladoTanqueMezclado2.mlx] [[PDF](#)] [\[YouTube ▶\]](#)

En este vídeo se modela un tanque de mezclado, similar al del vídeo [22], pero incorporando una dinámica adicional en el mezclado (concentración interior del tanque no homogénea).

La concentración no homogénea se modela dividiendo el volumen del depósito en dos mitades, que intercambian un caudal q_{12} entre ellos (los detalles de la determinación de la fórmula de dicho caudal se aborda en el vídeo [537]). Obviamente, el modelado más fiel sería simulando ecuaciones en derivadas parciales (Navier-Stokes, etc.) pero, vaya, eso complica mucho la simulación y en muchos casos se prefieren aproximaciones de orden

bajo (por ejemplo, las aproximaciones de primer orden de un intercambiador de calor en el vídeo [29]).

La parte fundamental de la dinámica se ha modelado en el Symbolic Toolbox: tras proponer las variables intervinientes y las ecuaciones que las relacionen, se analiza primero si el modelo está bien planteado como sistema de ecuaciones diferenciales y, a continuación, se obtendrá la representación interna normalizada mediante el comando `solve`.

Las ecuaciones de estado y de salida, las saturaciones, retardo de transporte y dinámica de válvulas se integrarán todas en bloques Simulink.

[539: tmrmod3] Modelo tanque de mezclado (v2): integración en Simulink, dinámica adicional, saturación y retardo

Materiales: [CÓD.: TanqueDeMezclado2S.slx] [PDF]

*** 10:26

[YouTube ▶]

En este vídeo se integran en un subsistema de Simulink las ecuaciones de estado y de salida que se dedujeron a partir de leyes de la Física en el vídeo [538].

Se incorpora, además, la saturación de determinadas entradas, el límite inferior (cero) de los volúmenes (umbrales en el integrador), dinámica de las válvulas, así como retardo de transporte (medida de concentración aguas abajo, con retardo inversamente proporcional al caudal).

[540: tmrsim] Tanque de mezclado no homogéneo: simulación

Materiales: [CÓD.: tanquemezcladoSimularLinealizar.slx] [PDF]

*** 05:45

[YouTube ▶]

En este vídeo se simula el subsistema codificado en Simulink en el vídeo [539], integrándolo en un modelo completo de Simulink donde las salidas se visualizan con bloques `scope` y las entradas son generadas por bloques `repeating sequence`. Se comprueba el efecto de la dinámica de concentración simulando la concentración superior, inferior y aguas abajo tras el retardo.

[541: tmrlin] Linealización tanque (V2) con dinámica de mezclado y retardo (Simulink)

Materiales: [CÓD.: LinealizarTanqueV2.zip]

*** 06:18


[YouTube ▶]

Este vídeo linealiza el modelo de tanque de mezclado con retardo de transporte discutido en los vídeos [538] y [539], usando el comando `linearize` de Simulink. El desarrollo es muy parecido al del vídeo [97], pero ahora se establecen diferentes opciones para asegurar que el resultado es un modelo en tiempo continuo y que *no* se aproxima el retardo mediante Padé, que sería la opción por defecto. Para comprobar el grado de no-linealidad, se linealiza en siete puntos de funcionamiento diferentes, y se observan las diferencias en la respuesta ante incrementos escalón de entradas / en frecuencia en los distintos puntos de operación.

El cuarto de los siete puntos de operación será el que se escoja como punto nominal para el diseño de controladores.

Diseño del control (lineal)

[542: timplant] Tanque de mezclado (V2): planteamiento del problema de control

***  06:59

Materiales: [MixingV2planteam.pdf]

[YouTube ▶]

En este vídeo se plantean los rasgos generales de la estructura de control a diseñar para el proceso de tanque de mezclado modelado en el vídeo [539], y linealizado en el vídeo [541].

Básicamente, dado que hay tres salidas (Q_s , $x_{a,out}$, h), y dos actuadores (V_{out} , q_B), se planteará un control en cascada con sensor extra de modo que Q_s sea controlado con V_{out} y se disponga de dos nuevas variables manipuladas (Q_b , ref_{Q_s}) para controlar $x_{a,out}$ y h , respectivamente, en una estrategia multibucle.


En el diseño de control se analizará la mejora de prestaciones si se incorporara desacoplamiento, y si se dispusieran de sensores de caudal de entrada q_A y de pureza del componente A ($x_{a,in}$) para hacer prealimentación (feedforward).

Tras el planteamiento del problema discutido aquí, el diseño en concreto y su simulación se abordan en otros vídeos de la colección, comenzando por el [543] y [544].

Otro caso de estudio de un problema de mezclado+calentamiento, resuelto por “sentido común” (sin modelos) se aborda en los vídeos [532] y [534], cuya visualización se aconseja. El mezclado estático se aborda en detalle en el vídeo [533].

Estructura básica: cascada + multibucle:

[543: tmcas] Tanque de mezclado (V2): diseño de bucle cascada esclavo de control de caudal

****  11:00

Materiales: [Cód.: DisenyoControlTanqueNominalEsclavoCasc.mlx] [PDF]

[YouTube ▶]

Este vídeo discute el diseño para el modelo de tanque de mezclado linealizado en el vídeo [541], del control en cascada (esclavo) de caudal, de modo que una vez cerrado el bucle de control en cascada, de modo que las variables manipuladas sean caudal de salida ref_{Q_s} y caudal q_B , según se había esbozado en el vídeo [542].

En pocas palabras, se alimenta al comando **pidtune** de la Control System Toolbox con el modelo entre V_{out} y Q_s , y se deja que diseñe un PI con las opciones por defecto, de modo similar al que un operador de planta activaría la opción de *autosintonía* de un PID industrial.

El objetivo del vídeo, no obstante, es analizar las características del modelo en bucle abierto y en bucle cerrado resultante, para comprender mejor lo que está sucediendo.

En efecto, el modelo del proceso tiene un *cero* en el origen: no se puede mantener un caudal de salida constante sin que la entrada (abertura de válvula) crezca hasta infinito. En el vídeo se explica la razón física de ello (hay que abrir más conforme el depósito se va vaciando), y el efecto de esta cancelación polo-cero en el bucle cerrado.

El análisis se hace tanto en tiempo (step) como en frecuencia (**bodemag**).

Una vez verificado el comportamiento del bucle cerrado, se genera un nuevo modelo **G2** cuyas entradas sean perturbaciones de caudal y composición en el *wild feed* y referencias de caudales de B en el *controlled feed* y de caudal de salida (al control aquí diseñado). Este modelo se usará en fases posteriores para multibucle, desacoplamiento y prealimentación.

Nota: en muchas aplicaciones, habría sido suficiente (y aconsejable) suponer que el controlador en cascada se comporta “suficientemente bien” para poder asumir que la variable manipulada es el caudal de salida, sin detallar el control interno ni cerrar el bucle con un modelo complejo del proceso... modelando directamente de una forma similar a la realizada en el vídeo [96], y eliminando los aspectos relacionados con la válvula de salida

del modelo: el objetivo “práctico” del control en cascada es *simplificar* y no *complicar* las cosas; el objetivo de los profesores de control es enseñar cosas difíciles (complicando, a veces, las cosas sin que estrictamente fuese necesario en la práctica).

[544: tmrml]
Tanque de mezclado (V2): control multibucle (con cascada preexistente)

*** 12:52

Materiales:

CÓD.:
[
tmmb.zip
]

[YouTube ▶]

Este vídeo aborda el diseño del control multibucle del tanque de mezclado cuyo modelo linealizado se obtuvo en el vídeo [541], suponiendo un bucle de control en cascada de caudal de salida preexistente (diseñado en el vídeo [543]), siguiendo los pasos a la solución global del problema esbozados en el vídeo [542].

El emparejamiento seleccionado es nivel \rightarrow caudal de salida, concentración \rightarrow caudal de disolvente q_b .

Dado que, en bucle cerrado, de referencia de caudal de salida a nivel se tiene integrador, se diseña con pidtune un regulador *proporcional* que consigue una dinámica satisfactoria ante referencia. No obstante, tiene error de posición ante perturbaciones en la entrada según se justifica cualitativamente y se verifica en simulación (un bucle cerrado estable con integradores debe hacer cero la entrada al integrador... que sólo es el error si el integrador está en el regulador). Por ello, al final del vídeo, se rehace el diseño y se sintoniza un regulador **PI**.

El control de concentración se hace, también con pidtune, con un **PI**.

Al final del vídeo, se usa el comando connect para simular el bucle cerrado resultante, tanto ante cambios de referencia como ante perturbaciones en respuesta ante escalón (el estudio del error en bucle cerrado en el dominio de la frecuencia se aborda en el vídeo [547]).

Refinamientos: prealimentación y desacoplamiento (lineal)

[545: tmff1]
Tanque de mezclado (V2): prealimentación

**** 09:47

Materiales:

CÓD.:
[
TanqueFFyDCStatic.zip
]
PDF
]

[YouTube ▶]

Nota: En la grabación del vídeo existía un error (bastante irrelevante) en lo que respecta a prealimentación/desacoplamiento en el modelo del control en cascada. Ha sido corregido en el archivo .zip adjunto de materiales.

Este vídeo discute el feedforward (en ganancia estática) para el modelo de tanque de mezclado linealizado en el vídeo [541], una vez cerrado el bucle de control en cascada, de modo que las variables manipuladas son caudal de salida ref_{Q_s} y caudal q_B . Los sensores para prealimentación serán el caudal y la pureza del reactivo entrante por la alimentación izquierda ($Q_a, x_{a,in}$). Obviamente, la decisión sobre si se compran o no dichos sensores podría venir aconsejada según la mejora de prestaciones con/sin prealimentación en simulaciones.

Sin asumir ningún desacoplamiento preexistente, la solución se obtiene con inversas de matrices (teoría en vídeo [477]). Como el proceso tiene integradores, debe calcularse la prealimentación con los valores de las matrices de ganancia a una frecuencia suficientemente pequeña pero no cero, y comprobar que cambiando dicha frecuencia no cambian apreciablemente las matrices obtenidas (esto es, que existe el límite cuando $s \rightarrow 0$ de las matrices de prealimentación). En este vídeo, se ha usado el comando evalfr para llevar a cabo este paso.

Se analiza el significado físico de la matriz de prealimentación, comparándola con el cálculo de nuevos puntos de funcionamiento constante ante los nuevos valores de las variables de

entrada no manipuladas: para pequeños incrementos, el resultado linealizado coincide con dicho cálculo de puntos de funcionamiento. Para incrementos grandes, se plantea la conveniencia de hacer un feedforward con los cálculos no lineales exactos como, de hecho, se hace en muchas aplicaciones mediante ratio control. En este caso de estudio, esto se realiza, por ejemplo, en el vídeo [550].

La prealimentación lineal asumiendo desacoplamiento preexistente, y la comparación con los resultados de este vídeo, se discuten en el vídeo [546].

[546: tmdcff2] Tanque de mezclado (V2): desacoplamiento y prealimentación
desacoplada

Materiales: [[Cód.:](#) TanqueFFyDCStatic.zip] [[PDF](#)]

**** 10:31

[YouTube ▶]

Este vídeo discute el desacoplamiento (en ganancia estática) para el modelo de tanque de mezclado linealizado en el vídeo [541], una vez cerrado el bucle de control en cascada, con variables manipuladas caudal de salida ref_{Q_s} y caudal q_B , que se supone controlan concentración de salida y nivel, respectivamente, en una estructura multibucle.

En efecto, aunque el control de nivel (Q_s) no influye sobre la concentración (en régimen permanente), el de concentración (q_B) sí influye sobre nivel. Por ello, un desacoplamiento que incremente en 1 unidad Q_s cuando se incrementa en la misma cantidad q_B podría ser aconsejable.

La segunda parte del vídeo discute la prealimentación con los sensores de caudal y pureza del reactivo entrante por la alimentación izquierda ($Q_a, x_{a,in}$).

Sin suponer desacoplamiento, las inversas de matrices asociadas se han discutido en el vídeo [545]. Con desacoplamiento preexistente, sólo hay que dividir por la *diagonal* de la matriz de transferencia, resultando, quizás, más sencillo de explicar a no expertos en control multivariable. Se analiza el significado físico de las diferentes ganancias de feedforward en el caso desacoplado, así como la relación con las del vídeo [545]. La extensión al modelo no lineal (en equilibrio) de estos cálculos con la matriz de ganancia se aborda en el vídeo [550].

[547: tmrffdyn] Tanque de mezclado (V2): prealimentación y desacoplo
dinámicos

Materiales: [[Cód.:](#) TanqueFFyDCDynamic.zip] [[PDF](#)]

**** 09:47

[YouTube ▶]

Este vídeo discute los aspectos dinámicos del diseño de prealimentación y desacoplamiento para el modelo de tanque de mezclado lineal (vídeo [541]). El diseño puramente estático (bloques feedforward y desacoplamiento constantes) ha sido discutido en los vídeos [545] y [546]... es importante destacar que la conclusión de este vídeo será que esos diseños constantes son muy razonables.

Los sensores para prealimentación eran el caudal y la pureza del reactivo entrante por la alimentación izquierda ($q_w, x_{a,in}$). Se supone preexistente un control en cascada diseñado en el vídeo [543].

En primer lugar, se hace un análisis sobre si prealimentación y desacoplo son realmente necesarios y hasta qué rapidez (ancho de banda), examinando el diagrama de Bode de magnitud (**bodemag**) de los errores ante perturbaciones y referencias. Se comprueba que con un ancho de banda de aproximadamente 10 rad/s sería suficiente.

Sin asumir ningún desacoplamiento preexistente, la solución de la prealimentación se obtiene con inversas de matrices (teoría en vídeo [477]); en este caso, con la inversa de *matrices de transferencia*. Dibujando su diagrama de Bode, se comprueba que no es realizable (respuesta tiende a infinito) pero que hasta 10 rad/s es aproximadamente constante, lo cual justifica la conveniencia de la solución *estática* antes mencionada.

En cuanto al desacoplamiento (en forma inversa), se obtiene un bloque con dinámica realizable (aunque, de igual modo, dado que no hay cambios relevantes hasta 10 rad/s, sería muy razonable implementar el desacoplamiento constante estático).

Simulación del sistema controlado resultante

[548: tmsimtodolin] Tanque de mezclado (V2): simulación de multibucle, cascada, desacoplamiento y prealimentación (modelo linealizado)
06:25

Materiales: [[CÓD.:](#) TMsimulinktodoconlineal.zip]

[YouTube ▶]

Este vídeo implementa en un diagrama de bloques de Simulink la estructura de control completa planteada en el vídeo [542], sobre el modelo linealizado. Incorpora:

- Modelo linealizado del tanque (4 entradas, 3 salidas), del vídeo [541],
- Control en cascada (esclavo) de caudal de salida, manipulando válvula de salida, vídeo [543],
- Control multibucle de nivel y concentración, vídeo [544], manipulando referencia a control de caudal de salida y caudal entrante de disolvente q_b , respectivamente.
- Prelimentación estática (matriz, vídeo [545]), con sensores de caudal y concentración del *wild feed*, y
- desacoplamiento (dinámico, vídeo [547]).

Mediante los adecuados bloques **Manual Switch**, se simulan distintas combinaciones de desacoplamiento o feedforward, bien presente o bien ausente.

[549: tmsimbcnl] Tanque de mezclado V2: simulación no lineal del control linealizado
Materiales: [[CÓD.:](#) simulaffydcNOlineal.slx]

*** 08:27

[YouTube ▶]

Este vídeo implementa en un diagrama de bloques de Simulink la estructura de control completa planteada en el vídeo [542], sobre el modelo linealizado, que se simuló en el vídeo [548] con dicho modelo.

En este vídeo se simula sobre el modelo no lineal construido en el vídeo [539], incorporando los adecuados pasos desde variables en unidades absolutas a incrementales y viceversa. Al igual que en el vídeo del modelo lineal, mediante los adecuados bloques **Manual Switch**, se simulan distintas combinaciones de desacoplamiento o feedforward presente o ausente.

Se observan los siguientes comportamientos que la simulación lineal no consideraba:

1. Dinámicas del transitorio diferentes si se cambia el punto de funcionamiento (referencias).
2. Efectos perjudiciales de la saturación (mala cancelación de perturbaciones extremas e inestabilidad interna –windup– de los reguladores PI cuyo actuador satura).

En el primer caso, podría plantearse una planificación de ganancia. En el segundo caso, debería tenerse en cuenta la presencia de saturación mediante antiwindup o/y cambios en el desacoplamiento y feedforward. Las soluciones a estos problemas *no* entran dentro de los objetivos de este vídeo.

Control no lineal (estático), antiwindup

[550: tm2ffnl] Tanque de mezclado: prealimentación, desacoplamiento, 2GL, ratio control con modelo no lineal
Materiales: [[CÓD.:](#) tm2ffnl.zip] [[PDF](#)]

**** 11:48

[YouTube ▶]


Este vídeo diseña, para el caso de estudio del tanque de mezclado, usando un modelo no lineal, una estructura que calcula el punto de equilibrio de las variables manipuladas q_b y ref_{q_s} en función de las perturbaciones y referencias. Con ello, se unifica el cálculo (en régimen permanente) de bloques que en control lineal serían prealimentación, control con 2 grados de libertad y desacoplamiento, según lo discutido desde un punto de vista genérico en el vídeo [526], cuya visualización se aconseja para poder extraer conclusiones generales importantes en ingeniería de control, más allá de este problema concreto. Este vídeo, por tanto, sería la generalización al caso no lineal (pero sólo en régimen permanente, equilibrio constante) de los cálculos lineales de prealimentación y desacoplamiento vistos en los vídeos [545] y [546].

El cálculo está dividido en dos bloques **Matlab Function** de Simulink:

- Cálculo del caudal de disolvente en equilibrio en función de q_w , $x_{a,in}$ y $ref_{x_{a,out}}$, equivalente a prealimentación + 2 GL,
- Cálculo del caudal de salida dados q_w y q_b (siendo éste determinado por el punto de funcionamiento y bucle de concentración). Esto sería equivalente a prealimentación y desacoplamiento dado que q_b puede no ser igual al que el primer cálculo haya decidido, debido a transitorios del bucle de concentración y posibles saturaciones.

[551: tm2aw] Tanque de mezclado no lineal: diseño del antiwindup

Materiales: [[CÓD.:](#) tm2ffnl.zip] [[PDF](#)]

***  08:22


[YouTube ▶]

Este vídeo discute la elección de las distintas estrategias de *antiwindup* en el control en cascada esclavo (se elige **clamping**), y en los PI de nivel y concentración (debido al cálculo de punto de funcionamiento externo, variable, y ser maestros de controles en cascada, se elige **tracking-mode antiwindup**). La teoría del antiwindup con entrada externa de seguimiento se discute en los vídeos [521] y [522], y la sintonía de la dinámica del tracking en el vídeo [523], cuya visualización previa se recomienda. El funcionamiento de la estrategia decidida se comprueba en simulación en la parte final del vídeo.

Control override

[552: tm2orm] Tanque de mezclado: necesidad de override para proteger de desbordamientos (motivación)

Materiales: [[CÓD.:](#) tmsolfinal.zip] [[PDF](#)]

**  03:24

[YouTube ▶]

Este vídeo simula la estructura de control completa diseñada en los vídeos [550] y [551] para un tanque de mezclado no lineal, ante fallos en la válvula de salida o referencias de concentración y nivel excesivamente bajas. Se observa que se pueden alcanzar niveles muy diferentes al punto de funcionamiento y se motiva, por tanto, la conveniencia de añadir un control “override” de modo que si el nivel sale de unos límites entonces el actuador q_b pase a controlar nivel en vez de concentración. Los detalles de ese diseño se ven en el vídeo [553].

[553: tm2orpp] Tanque de mezclado: override nivel mínimo y máximo (control proporcional)

Materiales: [[CÓD.:](#) tmsolfinal.zip] [[PDF](#)]

****  12:03


[YouTube ▶]

Este vídeo detalla la lógica de “*override*” en el tanque de mezclado para intentar no superar unas referencias de nivel máximo y mínimo. Básicamente, en situación normal el actuador q_b (caudal de disolvente) será usado para controlar concentración, pero si el control de nivel no cumple su objetivo (por saturación o fallo en la válvula de salida, o por una referencia de usuario fuera de límites, tal y como se simuló en el vídeo de motivación [552]) entonces se producirá el override y q_b pasará a controlar nivel.

Se proponen dos controladores de nivel proporcionales, y el vídeo discute las razones que justifican la propuesta de lógica de conmutación entre los tres posibles modos de control (overflow, underflow, normal). Se simula el resultado y se observa que existe un “error de posición” de modo que los límites del override son sólo aproximados. La incorporación de acción integral en los reguladores con override podría ser razonable, pero eso complica el escenario *antiwindup*; estas ideas se discuten en el vídeo [554].

[554: tm2orpi] Tanque de mezclado: override con acción integral y opciones
antiwindup

Materiales: [[CÓD.:](#) tmsolfinal.zip] [[PDF](#)]

****  10:34

[YouTube ▶]

En este vídeo se plantea usar reguladores PI para que cuando el override en el tanque de mezclado se active, que se alcancen las referencias de seguridad (aunque podrían ser brevemente rebasadas en el transitorio) debido a la acción integral (para evitar los problemas de error de posición discutidos en el vídeo [553] con override proporcional). Ello requiere pensar en diferentes opciones antiwindup (adicionales a las del bucle de concentración) y se prueban clamping y back-calculation (con distintas constantes K_t) así como PI con entrada externa de seguimiento (tracking mode) que, conceptualmente, sería la solución teóricamente mejor para seguir al controlador activo en vez de simplemente esperar en saturación a ser activado, aunque también tiene sus inconvenientes prácticos.

14.9.4 Otros ejemplos

A continuación, se listan enlaces (externos) a ejemplos adicionales de estructuras de control que combinan varios de los elementos explicados en este capítulo:


- Sistema de 3 depósitos (Autores: electronika14)
- Mezclado de dos componentes, y calentado para reacción. (Autores: electronika14)
- Proceso de combustión (Autores: controlguru.com)
- Aplicaciones del override y selector (Autores: electronika14)
- Aplicaciones del ratio control (Autores: electronika14)

14.10 RESUMEN: visión general del control ‘usual’ en industria

Esto seguramente no es “el control teórico que estudiarás en Doctorado”, pero si estás en un máster de “ingeniería” (industrial, automática, procesos), el resumen a continuación es importante “en la práctica”.

[555: pidbigpic0] Control industrial (descentralizado), el gran esquema de las cosas (1): Motivación, presentación

Materiales: [DecentrBigPicturev2ESP.pdf]

***  10:44

[YouTube ▶]


* [LINK TO ENGLISH VERSION](#)

Este vídeo presenta una serie de tres vídeos para hablar del ‘gran esquema de las cosas’ en ingeniería de control industrial. La audiencia objetivo de la serie de vídeos es un estudiante que ha dado ‘varios cursos de control’ seguramente interesado en especializarse en control avanzado de procesos... El énfasis en el ‘detalle’ en cada uno de los cursos puede hacer que ninguno de tus profesores te haya dado una ‘perspectiva global’ sobre qué se espera de un ‘ingeniero de procesos industriales’ o ‘ingeniero de control’ fuera de las aulas. Es difícil, pero hay que intentarlo y voy a poner mi granito de arena.

En la serie de dos vídeos (aparte de esta introducción) desgranaremos las siguientes ideas:

- Diseñar procesos fáciles de controlar (vídeo [556])
- Seleccionar la estructura de control más adecuada ([556])
- Sintonizar reguladores (vídeo [557])
- Llevar a cabo la implementación ([557])
- Tener un plan de análisis de datos para supervisión y mejora continua del proceso. ([557])

Los vídeos serán, intencionalmente, más largos de la cuenta porque el objetivo es tener ‘condensado’, ‘todo en uno’ la gran cantidad de opciones y las ideas clave de cada una de estas etapas, enumerándolas sin entrar en detalle. Obviamente, dicho detalle daría para varios cursos (y en efecto, así ocurre en las especialidades de grado y máster en control). Si alguna de las cosas aquí mencionadas no te suena, deberás buscar en otros materiales (de esta colección, quizás) dicho detalle en particular.

[556: pidbigpic1]	<p>Control descentralizado, gran esquema de las cosas (2): Proyecto de procesos, estructuras de control Materiales: [DecentrBigPicturev2ESP.pdf]</p>	<p>***  45:39 [YouTube ▶]</p>
-------------------	---	---

*[LINK TO ENGLISH VERSION](#)

Este vídeo detalla los conceptos siguientes:

- Diseñar/proyectar procesos fáciles de controlar
- Seleccionar la estructura de control más adecuada

del ‘gran esquema de las cosas’ (vídeo [555]) en control de sistemas complejos industriales, que continuará en el vídeo [557] con la discusión sobre:

- Sintonizar reguladores
- Llevar a cabo la implementación
- Tener un plan de análisis de datos para supervisión y mejora continua del proceso.

En concreto, se discute el significado de optimizar el punto de operación (el coste por unidad de producto, coste variable, depende de la ‘media’ de consumos), y de dimensionar actuadores (actuadores más potentes requieren mayor inversión de capital a amortizar como coste fijo).

Luego, se discute la conveniencia de seleccionar las variables controladas y actuadores con la tecnología adecuada para que la dinámica sea rápida y de alta ganancia, la conveniencia de tener cuantos más sensores mejor, y se enumeran los posibles defectos de diseño que harían el control difícil.


Una vez se tiene un proceso bien diseñado, se discuten aspectos variados de la estructura de control (flujo de información, división de tareas), en el caso *descentralizado* (no vamos a meter decenas o cientos de sensores y actuadores a un único computador, en principio).

En concreto, se discute:

- Configuracioens base multilazo, desacoplamiento

- Sensores extra (cascada, fusión sensorial, prealimentación)
- Sensores faltantes (lazo abierto, inferencial, observadores)
- Actuadores extra (fusión de actuadores, cascada)
- Sensores Y actuadores extra (gradual, referencias secundarias)
- Control jerárquico (cascada múltiple)
- Aspectos de seguridad en bucles (detección de fallos, override, redundancia)
- Refinamientos (bloques no lineales, medias, máximos, ...)

También se acaba diciendo que las cosas se hacen de cierta forma en ciertos sectores tecnológicos por alguna razón: conoce tu proceso y las razones de las decisiones que otros tomaron antes de decidir tú cambiar cosas.

[557: pidbigpic2]	Control descentralizado, gran esquema de las cosas (3): sintonizado, implementación, análisis de datos Materiales: [DecentrBigPicturev2ESP.pdf]	***  33:52 [YouTube ►]
-------------------	--	--

*[LINK TO ENGLISH VERSION](#)

Este vídeo concluye la presentación del gran esquema de las cosas en control industrial ‘de todos los días’ (instalaciones ultra-modernas y equipos de I+D podrían no estar alineadas con lo que aquí se comenta). Tras haber discutido en vídeos previos el diseño del proceso para que sea fácil de controlar y la selección de variables controladas, manipuladas, medidas y estructura de control, aquí discutimos el sintonizado de reguladores, la implementación y el análisis de datos.

El sintonizado de reguladores discute sobre si merece la pena o no sintonizar todos los bucles, sobre la importancia del control todo-nada, sobre la sintonía con/sin modelos (ajuste experimental en el segundo caso), y sobre algunos aspectos avanzados y refinamientos.

La discusión sobre implementación versa sobre herramientas software a usar, supervisión, bumpless transfer, seguridad, registro de eventos, etc.

El análisis de datos debe hacerse para planificar producción, mantenimiento, diseño de experimentos, y averiguar las causas físicas de la variabilidad: si la variabilidad se reduce, el punto de operación puede acercarse más a límites tecnológicos o de normativa, para conseguir ahorrar costes sin violar dichas restricciones.

Capítulo 15

Control centralizado: asignación de polos

Es obvio, pero conviene recordarlo que el concepto de “estado” de un sistema físico es clave en modelado (representación interna), pero también lo es en control. Se supone que el lector de este capítulo debe tener gran familiaridad con los conceptos de variables de estado en modelado de sistemas físicos (vídeos [estado1(10:48)] y [estado2(12:50)], abordando el marco conceptual) y saber cómo obtenerlos a partir de descripciones de los sistemas con ecuaciones de la física o identificación experimental.

15.1 Realimentación del estado

[558: OAaspolos]	Control por realimentación del estado: asignación de polos (introducción)	*** 	10:05
Materiales: [AsignPolos.pdf]			[YouTube ▶]

Se revisa el concepto de estado y la importancia de los polos para evaluar la dinámica de un sistema (estabilidad, tiempo de establecimiento, oscilaciones).

Se justifica que con el estado medible no es necesaria realimentación dinámica, sino que un regulador estático basado en el estado es suficiente.

Se explica, a partir de un ejemplo sencillo, cómo resolver el problema de diseñar un regulador $u = -Kx$ para que el bucle cerrado tenga los polos en una ubicación prefijada (asignación de polos).

[559: aspolos]	Asignación de polos (más largo)	*** 	18:16
Materiales: [Cód.: ejemploplace.mlx] [PDF]			[PLAY ▶]

Una versión un poco más larga del video [558] incorporando la resolución del ejemplo con Matlab. Los materiales teóricos son el mismo PDF que el citado vídeo. Aquí se consigna como material adicional el código Matlab ejecutado en este vídeo.

15.2 Estimación del estado: Observadores

[560: obs]	Estimación del estado: planteamiento del problema y concepto de “observador”	*** 	13:55
Materiales: [ObsIdeaBasica.pdf]			[YouTube ▶]

Este vídeo plantea la necesidad de estimar determinadas variables “importantes” (de estado) para la evolución futura de un sistema si no son directamente medibles. Presenta, pues, los conceptos asociados a dicho problema.

El vídeo revisa el concepto de variable de estado, y el sensor como medida imperfecta de dicho estado. Estos conceptos se introducen con mayor detalle en un contexto de modelado y teoría de sistemas en los vídeos [9] y [10].

Planteamiento del problema de estimación del estado; necesario punto intermedio entre modelo de ecuación de estado perfecto (sensores innecesarios) o modelo totalmente impreciso (estimar por mínimos cuadrados a partir de sensores).

Observador: diagrama de bloques.

La parte final enuncia brevemente posibles criterios de diseño de los observadores: estadístico (filtro de Kalman, vídeo [604]), determinista (fijar tiempo de establecimiento tras el cual el estado debe estar correctamente estimado, asignación de polos, vídeo [561]), u otros considerando criterios frecuenciales, error de modelado, etc. (por ejemplo el observador \mathcal{H}_∞ , vídeo [669]).

[561: obsap] Observadores por asignación de polos (teoría)

Materiales: [ObsAsignPolos2.pdf]

***  20:29
[YouTube ►]

Este vídeo aborda la estimación del estado (problema planteado en vídeo [560]) mediante la metodología de asignación de polos.

Primero presenta las ecuaciones básicas de un observador lineal: en tiempo continuo $\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x})$; en tiempo discreto $\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k - C\hat{x}_k)$ y lo que se conoce como observador discreto adelantado $\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k - C(A\hat{x}_k + Bu_k))$. La matriz L es el *parámetro de diseño*.

Con dichas ecuaciones se calcula la dinámica del error (suponiendo que el proceso real es $\dot{x} = Ax + Bu$, $y = Cx$ (o su versión discreta).

Dicha dinámica depende de $A - LC$ (o $A - LCA$ en el adelantado) y ello justifica calcular la matriz L de modo que los polos de dicha dinámica estén en posiciones prefijadas, lo que constituye la metodología de diseño de observadores por asignación de polos.

La parte final del vídeo presenta un breve ejemplo y discusión final.

El efecto del ruido sobre los procesos es importante, y se recomienda al lector visualizar el vídeo [599] para su análisis.

[562: obsml1] Observadores por asignación de polos (Matlab, ejemplo sencillo)

Materiales: [[Cód.:](#) ejemploplaceobss.mlx] [[PDF](#)]

***  17:21
[YouTube ►]

En este vídeo, se presenta un modelo de segundo orden, se discretiza y se diseña un observador (estimador del estado).

Se simula el proceso ante una entrada conocida, y se compara la salida del estado simulado con la del estimado calculado por el observador, comprobando que el error tiende a cero.

También se simula el efecto de un ruido, de modo que el error no tiende a cero, sino que solamente tiende a cero en media, con una cierta “varianza residual”. Una aproximación preliminar al efecto del ruido en los observadores se discute en el vídeo [599]; el observador que minimizaría dicha varianza residual sería el filtro de Kalman (vídeo [604]).

En el vídeo [567] se diseña otro observador y se obtiene su respuesta en frecuencia, para un doble integrador. También se compara con un observador de “orden reducido” (ver vídeo [563]).

15.2.1 Observador de orden reducido

[563: roobste] Observador lineal de orden reducido: teoría
Materiales: [obsordredT.pdf]

****  12:27
 [YouTube ▶]

Este video presenta el desarrollo teórico de lo que se conoce como observadores de orden reducido. En estos observadores se dispone de un sensor de una parte de los estados $y = x_1$, y se desea estimar un observador de sólo la parte de la dinámica (x_2) no medida directamente. Cualquier sistema lineal puede ser transformado a $y = x_1$, como se discute en el vídeo.

Un ejemplo Matlab de observadores de orden reducido se discute en los vídeos [567] y [568].

15.3 Realimentación de la salida

[564: sepc] Realimentación de la salida: Principio de separación (caso continuo)

Materiales: [PrincSepC.pdf]

***  10:51
 [YouTube ▶]


Al combinar un observador que estima el estado interno de un proceso y un controlador por realimentación de dicho estado estimado, se tiene lo que se denominan controladores por realimentación de *salida*.

Este vídeo detalla la teoría subyacente para el caso continuo y demuestra el “principio de separación”, esto es, que el diseño separado de una realimentación del estado $A - BK$ y de un observador $A - LC$ por asignación de polos hace que el regulador $d\hat{x}/dt = (A - BK - LC)\hat{x} + Ly$, $u = -K\hat{x}$, conectado en bucle cerrado con $\dot{x} = Ax + Bu$, $y = Cx$ sea estable y tenga sus polos en las posiciones diseñadas de controlador/observador.

Los últimos minutos del vídeo presentan código Matlab para ilustrar los conceptos teóricos.

[565: sepd] Realimentación de la salida: Principio de separación (discreto normal/adelantado)

Materiales: [PrincSepDISC.pdf]

****  17:09
 [YouTube ▶]

En este vídeo se desarrolla la teoría de realimentación de la salida y principio de separación para el caso de tiempo discreto, de forma totalmente paralela al desarrollo del vídeo [564]. en el caso continuo. Dado el paralelismo con el vídeo [564], el presente vídeo puede dejarse para un visionado posterior, si fuera necesario por limitación de tiempo.

Se desarrolla el principio de separación tanto para el observador normal (asignación de polos de $(A - LC)$) como para el observador adelantado/actualizado (asignación de polos de $(I - LC)A$): se presentan, para ambos observadores, las ecuaciones de estado del regulador discreto por realimentación de la salida asociado y, mediante un cambio de variable, se demuestra una estructura triangular por bloques de las ecuaciones en bucle cerrado. Con ello, los valores propios de una matriz triangular por bloques corresponden a los valores propios de cada uno de los bloques en la diagonal, y queda demostrado el teorema.

Los últimos minutos del vídeo presentan código Matlab y su salida que ilustra los conceptos teóricos explicados con anterioridad.

[566: ofml] Realimentación salida (obs+realim. Estado) por asignación de polos, ejemplo Matlab 2o orden discreto

*** 09:21

Materiales: [Cód.: ejemploutputfeedbacksimple.mlx] [PDF]

[YouTube ▶]

En este video, se completa el diseño de observadores en el video [562] con un controlador por realimentación del estado, y se simulan en Matlab de forma conjunta (realimentación sólo de las salidas de sensor medidas, no del estado completo).

El control por realimentación de la salida (observador+realimentación del estado) presentado en vídeos anteriores está diseñado para llevar al punto de funcionamiento (cero) el estado del sistema ante condiciones iniciales no nulas. Si existen perturbaciones o errores en el cálculo del punto de funcionamiento, podría no funcionar correctamente (ver vídeo [614] para un ejemplo). La solución (en algunos casos) requiere ampliar el modelo con modelos de las perturbaciones o desviaciones en entrada, ver vídeos [571], [572], [576].

Nota: La realimentación de la salida (observador + realimentación del estado) puede ser extendida con relativa facilidad en el caso discreto cuando los períodos de muestreo de sensores y actuadores no coinciden (pero todos los actuadores tienen el mismo período y todos los sensores también); esto se denomina control *bifrecuencia* y se desarrolla en la Sección 18.2. CE

15.4 Caso de estudio: sistema mecánico no lineal de cuarto orden

15.5 Doble integrador

[567: roobsml1] Observadores de orden completo/reducido: comparativa, caso estudio Matlab (1: diseño, respuesta en frecuencia)

**** 11:58

Materiales: [Cód.: observordenredML.mlx] [PDF]

[YouTube ▶]

Este vídeo diseña observadores de orden completo y de orden reducido para el sistema $1/s^2$ (doble integrador), modelando, por ejemplo, la entrada de fuerza a una masa, de la que se mide la posición.

La teoría de los observadores de orden completo se discute en el vídeo [561], y la de orden reducido en el vídeo [563].

La parte final del vídeo (opcional) analiza las matrices de transferencia y respuesta en frecuencia de los observadores, para interpretar mejor su significado físico.

La simulación temporal de estos observadores se aborda en el vídeo [568], continuación de éste.

[568: roobsml2] Observadores de orden completo/reducido: comparativa, caso estudio Matlab (2: respuesta temporal)

**** 14:51

Materiales: [Cód.: observordenredML.mlx] [PDF]

[YouTube ▶]

En este vídeo se simulan (**lsim**) los observadores de orden completo y de orden reducido diseñados en el vídeo [567].

Una primera simulación se hace sin ruido ni errores de modelado, para comprobar la convergencia exponencial del error de observación a cero, según predice la teoría. Posteriormente, se incorporan ruidos de medida y proceso y una no-linealidad en el actuador (zona muerta de fuerza), para evaluar las prestaciones de estos observadores en esa situación. Se observa que el observador de orden reducido es más sensible a ruido (estimando velocidad) que el de orden completo.

Nota: la simulación de ruidos en "tiempo continuo" debe ser entendida sólo "informalmente", aquí se discretiza a 0.02 segundos y se suman señales de distribución normal.

15.5.1 sistema mecánico no lineal de cuarto orden


[569: obsmec]

26:50

Control sistema mecánico orden 4 por asignación de polos (I):

Modelado, observadores

*** / ****



Materiales:

[[CÓD.:](#) ejplaceobs2masas.mlx] [[PDF](#)]

[YouTube ▶]


Sobre el sistema mecánico de orden 4 del video [88], se revisa su modelado y propiedades (video [275]). Se diseña un observador lineal para el mismo y se comprueba su funcionamiento.

Dado que el sistema es no-lineal, se modifica el “predictor” (estimado en bucle abierto del siguiente estado) usando el modelo no lineal para que la exactitud del estimado sea mayor; ideas refinadas sobre estimación del estado en sistemas no lineales aparecen en los videos [616], [628], ...

[570: ofmec]

Control sistema mecánico orden 4 por asignación de polos (II):

realim. Salida + antiwindup



18:26

Materiales:

[[CÓD.:](#) ejoutputfeedbackplace2masas.mlx] [[PDF](#)]

[YouTube ▶]

En este video se incorpora la realimentación del estado al código Matlab del video [569] anterior. El controlador es lineal, pero el observador es no lineal. Se incorpora, asimismo, el conocimiento de los límites de saturación del actuador para que el observador no estime un estado con una entrada incorrecta (equivalente al anti-windup en otros tipos de reguladores, ver videos [673] y [674] para un análisis teórico en mayor profundidad).

Otro ejemplo (más complicado) de implementación de realimentación de la salida y anti-windup en un control con acción integral (que es donde realmente se produce el windup, cuando los integradores están en el regulador) se aborda en el video [581].

15.6 Rechazo de perturbaciones

[571: rp1]

Rechazo de perturbaciones: planteamiento del problema



10:59

Materiales:

[[CancelPertGeneralidades.pdf](#)]

[YouTube ▶]

Este video plantea el problema general de rechazo de perturbaciones y sus variaciones según la perturbación sea medible o no, se tenga modelo o no, o sea ruido de proceso / de medida / referencia.

Se plantean algunos ejemplos de problema a resolver.

Según los objetivos de control (tiempo de establecimiento / mínima varianza / comportamiento “peor caso”), un problema de rechazo de perturbaciones sobre un mismo proceso se resolverá de forma diferente (asignación de polos, \mathcal{H}_2 , \mathcal{H}_∞).

Una mayor abstracción de los problemas aquí considerados dará lugar a la metodología de “planta generalizada” discutida en el video [639].

15.6.1 Perturbaciones constantes: acción integral


[572: drconst] Rechazo de perturbaciones constantes (acción integral) mediante asignación de polos
Materiales: [CancelConst.pdf]

***  18:17
 [YouTube ►]

Este vídeo plantea el problema de rechazo de perturbaciones *constantes* mediante asignación de polos.

- Modelado: seguimiento de referencias constantes sin error, compensación de errores en punto de funcionamiento, deriva en sensores, etc.
- Cálculo de nuevo punto de funcionamiento a partir de ecuaciones en equilibrio.
- Realimentación del estado sobre nuevas variables incrementales.
- Si la perturbación es desconocida, modelo ampliado con integrador y cálculo de observador.
- Realimentación de la salida.

[573: drconstuml] Rechazo de perturbaciones constantes a la entrada (asignación de polos, Matlab)
Materiales: [[CÓD.:](#) ejemplodistrejcte.mlx] [[PDF](#)]

***  09:59
 [YouTube ►]


Este vídeo utiliza la teoría del vídeo [572] en un ejemplo Matlab donde se diseña un regulador que cancela una perturbación $\dot{x} = Ax + B(u + d)$, siendo d aproximadamente constante (error en cálculo de punto de funcionamiento). El ejemplo es un modelo masa-muelle-amortiguador de orden 2.

Se diseña una realimentación del estado $u = -K\hat{x}$ por asignación de polos de orden 2, se le añade $-\hat{d}$, esto es $u = -K\hat{x} - \hat{d}$, y se estima \hat{d} con un observador, siendo éste diseñado sobre un modelo ampliado con un integrador (de orden tres).

El controlador resultante, de dimensiones 1×1 (un sensor, un actuador), tiene un integrador en su función de transferencia, como intuitivamente se preveía.

Un ejemplo similar en tiempo discreto se aborda en el vídeo [575].

[574: drconstrml] Seguimiento de referencias constantes sin error (asignación de polos, ejemplo Matlab)
Materiales: [[CÓD.:](#) ejemplodistrejtc1gl.mlx] [[PDF](#)]


***  10:07
 [YouTube ►]

Este vídeo presenta un segundo ejemplo con Matlab. Utiliza la teoría del vídeo [572] sobre el mismo proceso que el ejemplo del vídeo [573]. En este caso, la perturbación tiene un efecto sólo sobre la salida, esto es $e = Cx - r$, y se desea llevar el error a cero cuando r es una constante. La estructura de control es de “1 grado de libertad”, esto es, se mide el error únicamente (el controlador no dispone de sensores y referencia por separado).

El código para diseño, cambiando un par de matrices E , F , es prácticamente igual al del caso del vídeo [573]. El controlador resultante $u = Reg(s) * e$ tiene, obviamente, un integrador. En estos ejemplos que realmente son de control monovariable (1 entrada, 1 salida), el resultado es básicamente igual que el del vídeo [573] aunque se parta de un marco conceptual diferente.

Otro ejemplo parecido aparece a continuación, en tiempo discreto:

[575: ofexi] Control discreto por realimentación de la salida con/sin acción integral, análisis en bucle cerrado (Matlab)
Materiales: [[CÓD.:](#) RealSalConSinIntegral.mlx] [[PDF](#)]

****  14:45
 [YouTube ►]

Este video compara los diseños basados en observador + realim. estado con/sin acción integral, mediante un modelo ampliado de perturbación a la entrada. Lo hace en un ejemplo Matlab discreto de proceso de orden 2 poco amortiguado, usando “observador adelantado” (current observer).

Las ganancias se calculan con `dlqr` + `dlqe` o bien con `place`. Mediante las adecuadas fórmulas de regulador por realimentación de la salida a partir de las K y L de observador adelantado, se construyen las representaciones internas de dichos reguladores y se simulan en bucle cerrado ante perturbaciones a la entrada y ante perturbaciones a la salida (entendidas como cambio de referencia) para analizar el comportamiento en bucle cerrado.

La “implementación” en control por computador del regulador diseñado (bueno, su simulación en un código que se parecería más al código final en una aplicación), se discute en el vídeo [581], continuación de éste.

El diseño \mathcal{H}_∞ y su simulación para este mismo proceso discreto se abordan en los vídeos [675] y [676].

15.6.2 Caso general: perturbaciones deterministas constantes, rampa, senoidales

[576: drgen]

Asignación de polos: rechazo de perturbaciones deterministas
(escalón, rampa, senoidal)

Materiales: [CancelDeterminist.pdf]

**** ☰ 19:20

[YouTube ▶]

En este vídeo se plantea el rechazo a cualquier perturbación determinista generada por un modelo (usualmente, marginalmente estable) $\dot{\xi} = M\xi$. Estos generadores, según M , pueden generar constantes, rampas, polinomios, senoidales (o combinaciones lineales de las mismas).

La idea básica es generalizar las ecuaciones de Sylvester que aparecen en el régimen estacionario de un sistema en representación interna (que se detallan en el vídeo [165]); se recomienda el visionado previo de ese vídeo para entender correctamente este. En el vídeo [165] se despejaba estado a partir de entrada. En este vídeo se despejan estado y entrada a partir de una perturbación tales que cancelen el efecto sobre unas salidas controladas.

El caso constante discutido en el vídeo [572] puede considerarse en un caso particular de los desarrollos aquí presentados.

Una vez resueltas las ecuaciones en régimen estacionario, el diseño de asignación de polos (modelo reducido para control, modelo ampliado para observador) es idéntico al caso constante (vídeo [572]).

Nota: por simplicidad, no se separan las salidas entre variables controladas / medidas, pero si ese fuera el caso, se aplicaría la cancelación sobre variables controladas y se diseñaría el observador a partir de las variables medidas, como se plantea en el vídeo [572].

[577: drgenml]

Asignación de polos: rechazo de perturbaciones deterministas
genéricas (ejemplo Matlab)

Materiales: [Cód.: ejemplodistrejslides.mlx] [PDF]

**** ☰ 14:08

[YouTube ▶]

Este vídeo presenta código Matlab que implementa los desarrollos teóricos del vídeo [576].

El proceso ejemplo es el mismo que en los vídeos [573] y [574]. La perturbación es colineal con la entrada $B = F$, como en el vídeo [573].

Se presenta una primera variación, particularizando las ecuaciones estacionarias al caso constante obteniendo, por tanto, una implementación equivalente al vídeo [573] que consideraba exactamente este caso particular.

Leugo, cambiando la matriz M del generador de ondas, se presenta otro ejemplo que diseña un regulador que cancela perturbaciones no medibles de frecuencia ω conocida.

Los resultados comparan: (i) los reguladores con modelos ampliados (que cancelan las constantes/senoides) de forma exacta, con (ii) los reguladores basados en modelo reducido (sin generador de perturbaciones), que tienen error de posición o vibraciones residuales.

Nota: Los filtros de Kalman permiten observar y cancelar perturbaciones basado en ideas de control óptimo en el Capítulo 16. En concreto, se aconseja al lector observar las analogías con los modelos de perturbaciones y observadores propuestos en los vídeos [rampdetr(07:57)] y [rampkales(12:38)].

Caso de estudio

[578: ceofp1]

Caso estudio realimentación salida asign. polos (1): modelado, discretización, selección especificaciones

10:58

Materiales:

[Cód.: ceofp123.zip]
[PDF]

[YouTube ▶]

Este es el primero de tres vídeos que describe un caso de estudio de control por realimentación de la salida de un sistema modelado como matriz de transferencia (continua) con elementos de tipo 2o orden+retardo.

En este vídeo se discute la selección de tiempo de establecimiento deseado, la discretización, la reducción de orden (balred) y la posición de los polos de controlador y observador a asignar (donde los polos de bucle abierto con dinámica más rápida de la deseada no son modificados, para obtener unas constantes de realimentación más pequeñas y “naturales” que tolerarán mejor los ruidos y errores de modelado presentes en todas las aplicaciones).

Nota: El hecho de elegir place en este material es para ilustrar cómo ejecutar dicha metodología, heredera de la “tradición” monovariable de mover polos en lugar de las raíces, por ejemplo; otras metodologías de control óptimo pueden ser más aconsejables en la práctica en cuanto a prestaciones y tolerancia a error de modelado. Por ejemplo, dlqr es otra opción quizás más aconsejable, dado que permite ajustar rapidez/amplitud de cada entrada y salida, mientras que la asignación de polos está muy limitada en ese aspecto, así como el filtro de Kalman dlqe como observador de mínima varianza dada una relación señal-ruido.

Los vídeos [579] y [580] continúan desarrollando el caso de estudio introducido aquí.

[579: ceofp2]

Caso estudio realimentación de la salida asign. polos (2): diseño de observadores/reguladores con/sin acción integral

11:30

Materiales:

[Cód.: ceofp123.zip]
[PDF]

[YouTube ▶]

Este vídeo es continuación del vídeo [578], donde se discretizó, aproximó y se eligieron polos para un proceso de 2 entradas y 2 salidas (resumido en el primer minuto y medio).


En este vídeo se ejecutan los comandos place y se construyen los reguladores por realimentación de la salida (entrada y , salida u) basados en observador “no adelantado” $\xi_{k+1} = A\xi_k + Bu_k + L(y_k - (C\xi_k + Du_k))$. En concreto, se discuten tres estrategias:

- Sin acción integral
- Con integradores como perturbación constante a la entrada observable pero no controlable
- Con integradores del error (error acumulado) a la salida, controlables pero no observables.

Se construyen todos los elementos necesarios para su implementación que será simulada en Simulink en el vídeo [580], continuación de éste.

[580: ceofp3] **Caso estudio realimentación de la salida asign. polos (3): simulación (Simulink)**

Materiales: [[CÓD.: ceofp123.zip](#)] [[PDF](#)]

****  13:34

[YouTube ▶]


Este vídeo es el tercero de un caso de estudio previamente analizado en detalle en los vídeos [578] (modelado, discretización, selección de los polos deseado) y [579] (construcción de observadores y controladores por asignación de polos con/sin acción integral). Los dos primeros minutos aquí resumen las conclusiones de dichos vídeos.

Este vídeo simula en Simulink los controladores sintetizados ante cambios de referencia y ante perturbaciones (ruido de proceso y de medida), presentando diagramas de Simulink que incorporan:

- El modelo “original” continuo de orden elevado y retardo (diferente a la aproximación de orden 4 que ha servido para diseño del control),
- El cálculo de punto de funcionamiento de entrada y salidas para que los observadores/controladores trabajen en variables incrementales (sin error de posición ante referencias, haya o no acción integral, si el modelo es correcto),
- Las modificaciones a los bucles necesarias para implementar acción integral comprobando que no hay error ante perturbaciones constantes no medibles.

[581: ofexi2] **Control discreto por realimentación de salida con acción integral: código de simulación y antiwindup**

Materiales: [[CÓD.: AWSStestOBS.mlx](#)] [[PDF](#)]

****  20:12

[YouTube ▶]

Este vídeo discute la implementación (en bucle de simulación) de un regulador con observador adelantado + realimentación del estado (o sea, “realimentación de la salida” en los libros de teoría). El detalle de cómo calcular las ganancias de realimentación del estado y de observador se discute en el vídeo [575], que se resume en los primeros dos minutos pero que se recomienda visualizar antes que éste.

En este vídeo se plantean dos alternativas de implementación del controlador:

- Con un observador explícito (predictor, corrector, realim. estado)
- Como un regulador lineal genérico una vez obtenida su representación interna.

La simulación muestra que los resultados son, obviamente, idénticos.

También se discuten las modificaciones para alcanzar un punto de funcionamiento diferente a cero (cambios unidades absolutas↔incrementales). En este vídeo se simulan los problemas de “windup” que aparecerían si se implementara el modelo lineal del regulador con acción integral sin más: hay que alimentar al predictor con la acción de control (incremental) que realmente recibe el proceso controlado; en la segunda implementación, hay que corregir el error multiplicando por la “B” del proceso la diferencia entre la acción de control ideal sin saturar y la acción saturada. Detalles teóricos sobre el antiwindup en reguladores genéricos no PID se abordan en los vídeos [673] y [674], que se aconseja visualizar.

El diseño \mathcal{H}_∞ y su simulación/implementación para este mismo proceso discreto se abordan en los vídeos [675] y [676]. La implementación de esos reguladores es, básicamente, idéntica a la segunda de las opciones aquí considerada, excepto el modo de calcular el antiwindup.

Capítulo 16

Control óptimo

16.1 Introducción al control basado en optimización

[582: optisa] Optimización: aplicaciones en ingeniería de sistemas y control, generalidades

**  10:34

Materiales: [IntroContrOptimo.pdf]

[YouTube ►]

En este vídeo se revisa en qué consiste la optimización (minimizar un cierto $J(\theta)$), y la optimización dinámica (si θ es una trayectoria, función del tiempo). Se presentan, desde una perspectiva muy general, sin apenas detalle matemático, las aplicaciones de los problemas de optimización en el entorno de ingeniería de sistemas (identificación experimental, optimización de parámetros físicos de procesos, optimización de puntos de operación) y de control óptimo (sintonizado óptimo de reguladores tipo PID, trayectorias de control óptimo de mínima energía, control predictivo, ...).

[583: coptidx] Control óptimo: índices de coste más usuales

***  12:11

Materiales: [IndiceCosteContrOptimo.pdf]

[YouTube ►]

Este vídeo presenta los índices de coste que se suelen usar en la mayoría de las aplicaciones de control óptimo: la integral del error al cuadrado (ISE) – ejemplo en vídeo [585]–, integral del valor absoluto del error (IAE) –ejemplo en vídeo [588]–, y las versiones con ponderación en el tiempo de los mismos. También se discute su versión en tiempo discreto (control por computador), reemplazando integrales por simples sumas. En el caso multivariable deben, obviamente, aparecer sumas ponderadas según los requisitos de rapidez o precisión de cada una de las variables controladas.

También es recomendable incluir en el índice de coste una ponderación de la acción de control, para evitar controladores de ganancia excesiva que den lugar a gran amplificación de ruido de medida, actividad indeseable de alta frecuencia, etc. (esto se hace en el ejemplo del vídeo [586] o, acotando directamente la actividad de alta frecuencia del control en el vídeo [588]). Se puede también incluir términos que dependan de valores máximos de salidas o acción de control; por ejemplo, el vídeo [789] plantea minimizar una combinación del IAE y del valor de pico para limitar sobreoscilación.

En resumen, en control óptimo, determinar una función objetivo (índice de coste) a minimizar que traduzca adecuadamente los requerimientos “prácticos” de un problema de control al lenguaje de la optimización es **el elemento más importante** de la metodología. Muchas veces el índice de coste se modifica por prueba y error hasta que se obtiene un bucle cerrado “satisfactorio en la práctica” evaluando no sólo el índice numérico sino las características de la respuesta temporal (pico, tiempo de establecimiento) y respuesta en

frecuencia (resonancia, amplificación de ruido de alta frecuencia, etc.) obtenidas por el regulador óptimo. Un ejemplo de la influencia de la elección del índice de coste en las características de la respuesta temporal del regulador óptimo aparece en los vídeos [590] y [591], cuyo visionado se recomienda para comprender en un ejemplo concreto las ideas genéricas que se plantean aquí.

Ejemplo/discusión sobre control de temperaturas de un horno:

[584: lsfor]

Mínimos cuadrados en control multivariable: caso estudio horno

(revisión/motivación)

Materiales: [[CÓD.:](#) HornoLSexamples.mlx] [[PDF](#)]

** 28:04

[[YouTube](#) ►]

Este vídeo hace una revisión de la utilidad de los mínimos cuadrados en control multivariable. Realmente es una excusa para revisar la interpretación de la matriz pseudoinversa, en un problema de un horno con cinco termopares y dos quemadores cuya acción se concentra en cada uno de los lados izquierdo y derecho. Se parte de una matriz de ganancia estática, supuestamente obtenida a partir de modelado teórico o identificación experimental, y se plantea qué se podría hacer para tener el horno ‘bajo control’, dado que con dos variables manipuladas es imposible controlar siguiendo referencias arbitrarias a un total de cinco variables controladas. Por ello se proponen cuatro opciones:

1. Intentar seguir cinco referencias ajustando por ‘mínimos cuadrados’, pseudoinversa de matriz con más filas que columnas.
2. Seleccionar dos temperaturas como variables controladas y desechar tres de ellas, rezando para que cuando las dos seleccionadas estén en su valor de referencia, que el resto estén ‘razonablemente bien’ para mi aplicación concreta.
3. Modular entre las dos opciones anteriores con mínimos cuadrados *ponderados*.
4. Controlar únicamente la temperatura media. En este caso, reducimos a “una” las variables controladas y, por tanto, con dos actuadores, existen infinitas combinaciones que dan la misma temperatura media. Escogeremos la solución mínimo cuadrática que hace $u_{left}^2 + u_{right}^2$ mínimo, mediante la “otra” pseudoinversa (matriz con más columnas que filas). Eso reparte la potencia de los actuadores proporcional a la ganancia de cada uno de ellos sobre la temperatura media.

Con ello, se han explorado las dos posibles concepciones de mínimos cuadrados (pseudoinversa izquierda o derecha), y su posible aplicación en un problema de control. En este caso, todas las opciones daban perfiles de temperaturas parecidos, pero podría no ser el caso ante perturbaciones o en otras aplicaciones con procesos peor condicionados. Esa discusión no es objeto del presente vídeo.

16.2 Sintonizado de reguladores PID basado en optimización

16.2.1 Simulink Response Optimization

[585: sdopid1]

Sintonizado de PIDs mediante optimización de integral error

cuadrático con Simulink Response Optimization

Materiales: [[CÓD.:](#) PIDTuneISEa.slx] [[PDF](#)]

*** 08:32

[[YouTube](#) ►]


En este vídeo se describe cómo utilizar el Simulink Response Optimization para ajustar los parámetros K_p , K_i , K_d de un PID de modo que se minimice la integral del cuadrado del error (integral square error, ISE) introducida en el vídeo [583].

Se ilustra cómo dicha integral a minimizar se introduce como un “requirement” del problema, y cómo las constantes del regulador se seleccionan para constituir el conjunto de “design variables”.

Una vez obtenido el controlador óptimo, el proceso en bucle cerrado presenta una respuesta ante referencia muy rápida, pero se observa una actividad de media-alta frecuencia en la salida y, especialmente, en la acción de control, excesivamente oscilatoria. Esto hace que, aunque sea *teóricamente* “óptimo” en cuanto a error cuadrático acumulado, posiblemente no sea una solución aconsejable desde un punto de vista *práctico*. El vídeo [586] discute las modificaciones al índice de coste para evitar estos problemas que puede original una excesiva actividad de la acción de control.

[586: sdopid2] Sintonizado de PIDs con Simulink Response Optimization:
ponderación ISE + acción de control

Materiales: [[CÓD.:](#) PIDTuneISEc.slx] [[PDF](#)]

***  08:02

[YouTube ▶]


Este vídeo es continuación del [585], donde se describía cómo utilizar el Simulink Response Optimization para ajustar los parámetros de un PID de modo que se minimizaba la integral del cuadrado del error (integral square error, ISE). En este vídeo se introduce un requerimiento adicional sobre la integral del cuadrado de la acción de control para conseguir bucles cerrados con menos actividad y oscilaciones de la acción de control, al modificar el índice objetivo de la optimización. Todo se realiza en el entorno de la herramienta de Response Optimization de Simulink.

Si la ponderación de la acción de control es muy grande, se genera un error de posición importante porque el optimizador “desactiva” la acción integral. En la práctica, salvo por cuestiones de saturación, lo que se desea limitar es la actividad de “alta frecuencia” del actuador (los “incrementos”, en reguladores en tiempo discreto). En el ejemplo aquí discutido, se propone realizar la integral del cuadrado de la salida de la acción de control tras pasar por un filtro paso-alto de primer orden (si el lector no tiene familiaridad con los filtros sencillos, puede visionar el vídeo [195]).

Con este tipo de poderaciones de la acción de control (o de parte de su espectro en frecuencia), el diseñador puede ajustar mejor el compromiso entre, por un lado, la rapidez y precisión del seguimiento de referencias y, por otro lado, la actividad de alta amplitud y frecuencia en el actuador que excitaría no-linealidades o dinámica no modelada (si esta actividad fuera excesiva, esto haría que el resultado en la práctica fuera demasiado poco tolerante a errores de modelado o a perturbaciones de ancho de banda elevado). El control teóricamente “óptimo” lo es desde un punto de vista numérico/matemático, pero que sea bueno en una aplicación práctica depende de que el índice de coste a optimizar esté especificado de forma adecuada. También pueden refinarse las especificaciones deseadas de la respuesta temporal o en frecuencia introduciendo restricciones en la optimización, como se discute en el vídeo [587].

[587: coptcn] Sintonizado de controladores con Simulink Response Optimization: restricciones sobre respuesta temporal

Materiales: [[CÓD.:](#) PIDTuneConstraints.slx] [[PDF](#)]

***  05:54

[YouTube ▶]

Este vídeo es continuación del [586]. En dicho vídeo se minimiza una suma ponderada de la integral del error cuadrático y la integral de la acción de control (con filtro paso alto).

Este vídeo introduce un retardo en el proceso, con lo que las prestaciones conseguibles disminuyen y el controlador inicial (todas las constantes del PID iguales a 1) no es estable (presenta un “ciclo límite” debido a la saturación). Si el retardo fuera grande (y conocido) podría pensar en integrarse un predictor de Smith en la estructura de control (vídeo [401]), pero eso no se ha abordado aquí, por simplicidad (aparte de que si el retardo es pequeño comparado con el tiempo de establecimiento objetivo, dicho predictor aporta complejidad

de implementación pero pocas ventajas en prestaciones respecto a la optimización aquí realizada).


La primera parte ejecuta rápidamente una optimización idéntica a la del vídeo [586], y en la segunda parte se introduce la restricción adicional de que la respuesta ante escalón ha de estar dentro de una cierta “envolvente” (Step Response Envelope), para exigir un tiempo de subida, tiempo de establecimiento y sobreoscilación prefijados.

Simulink Response Optimization realiza la optimización con restricciones y obtiene un bucle cerrado con la forma requerida de la respuesta ante escalón pero con un valor del índice de coste cuadrático mayor que cuando no existía la restricción (como era de esperar dado que el resultado con restricciones siempre será peor o como mucho igual al resultado de la optimización libre de restricciones).

La introducción de restricciones permite flexibilizar las opciones de diseño y establecer parámetros de la respuesta temporal que no tienen una relación demasiado directa con los índices ISE/IAE, filtros de acción de control, etc.

[588:sdofreq] **Sintonizado de controladores con Simulink Response Optimization: restricciones sobre respuesta en frecuencia linealizada**

Materiales: [[CÓD.:](#) PIDTuneConstraintsEj2.slx] [[PDF](#)]

****  09:26

[[YouTube](#) ►]

Este vídeo presenta un sistema de segundo orden con retardo y saturación de actuador, el cual se desea controlar con un regulador PID en la forma $u = K_P e + k_I \frac{1}{s} e + K_D \frac{N}{1+N/s} y$, de modo que no se deriva la referencia para evitar el “derivative kick” (incremento repentino hasta saturación con cambios en escalón de la referencia, ver <https://controlguru.com/pid-control-and-derivative-on-measurement/>). Asimismo se incorpora un antiwindup por tracking mode. Este PID se implementa con el bloque **PID Controller (2DOF)** de Simulink (otras consideraciones y una exposición con más profundidad de la idea del control con dos grados de libertad se aborda en los vídeos [517] y [518], y la estrategia de antiwindup se discute con algo más de detalle en el vídeo [469]).

El objetivo de todo esto es comprobar como Simulink puede optimizar los parámetros en cualquier estructura de bucle de control. En este caso el objetivo es minimizar la integral del valor absoluto del error (IAE, Integral Absolute Error, introducido en el vídeo [583]). En una primera optimización, se hace sin ninguna restricción ni modificación del índice de coste para limitar la acción de control (como se hacía, por ejemplo, en el vídeo [586]). El resultado es aceptable, pero se desea comprobar si la amplificación del ruido de medida de alta frecuencia es suficientemente baja y, si no lo es, introducir una limitación en la misma (Bode Amplitude constraint).

Como existen no-linealidades (saturación), Simulink debe linealizar para realizar esta comprobación. Para ello, se introducen puntos de “entrada” y “salida” para linealizar en lugares adecuados del diagrama de bloques (la entrada de la linealización será la variable de proceso que se introduce al PID, la salida de linealización será la acción de control). Se introduce un “requirement” sobre el diagrama de Bode de amplitud y se evalúa para comprobar dicha respuesta en frecuencia, y como no se considera satisfactoria, se edita el “requirement” para bajar la cota en dB máxima de la misma. Tras repetir la optimización, el resultado es un nuevo conjunto de parámetros del PID con menor acción derivada para amplificar menos el ruido, y obviamente resulta un IAE ligeramente mayor que el obtenido antes sin exigir las restricciones en frecuencia.

Nota: La idea de limitar el pico de la respuesta en frecuencia de diferentes funciones de un bucle cerrado es la motivación principal de la técnica de optimización \mathcal{H}_∞ para diseño de controladores (vídeo [639] y siguientes).

Rechazo de perturbaciones

[589: sdoPERT] Optimización para rechazo de perturbaciones vs. seguimiento de referencia (Simulink) **** 09:58

Materiales: [CÓD.: PIDTuneDisturbances19b.slx] [PDF] [YouTube ►]

Este vídeo sintoniza un PID para minimizar un índice combinación del IAE y de la integral del cuadrado de los componentes por encima de 0.4 rad/s de la entrada. A continuación, comprueba si dicho regulador cancela bien perturbaciones a la entrada de tipo escalón + “chirp” (barrido de frecuencias de 0.15 a 1.5 rad/s). Esta respuesta se compara con el resultado de optimizar el mismo índice de coste pero simulando ante dicha perturbación: obviamente, la nueva optimización consigue rebajar el valor del índice de coste a una tercera parte del obtenido con los parámetros óptimos ante referencia. Por tanto, es mejor optimizar ante los perfiles de entradas “previsibles” en cada aplicación. Se observa que la respuesta ante escalón en referencia del regulador óptimo ante perturbaciones tiene una oscilación poco amortiguada de alta frecuencia (que no estaba entre las frecuencias del barrido); eso lo coloca al borde de la inestabilidad pese a ser “óptimo”. La parte final del vídeo menciona la opción del control con 2 grados de libertad ($u = C_r(s)r + C_y(r)y$) como un enfoque posible para conseguir parámetros adicionales que permitan mayor flexibilidad en el diseño, se remite al lector al material [517] y [518] para más detalles.

Nota: Las resonancias elevadas podrían evitarse excitando dicha frecuencia en el barrido, añadiendo restricciones sobre la respuesta en frecuencia (vídeo [588]) de determinadas funciones de bucle cerrado, sobre el pico de la misma (lo que se denomina norma \mathcal{H}_∞ , ver `sdo.requirements.ClosedLoopPeakGain`) o con restricciones sobre la posición de los polos (`sdo.requirements.PZSettlingTime`, `sdo.requirements.PZDampingRatio`, ...). Por brevedad, se deja al lector que profundice y haga pruebas sobre el tema consultando la documentación de Matlab <https://es.mathworks.com/help/slido/specify-design-requirements.html>

Comparación de efecto de índices de coste en respuesta temporal

[590: opt4idxa] Optimización ISE/IAE/ITAE/IT2AE para rechazo de perturbaciones: planteamiento del problema *** 09:35

X. Blasco, J.M. Herrero, C. Ramos, A. Sala (UPV)

Materiales: [CÓD.: EjemploPI4IndicesFR.zip] [PDF] [PLAY ►]

En este vídeo se aborda la comparación de la respuesta temporal resultante de optimizar cuatro índices diferentes (ISE, IAE, ITAE, IT2AE) para un mismo proceso (1er orden + retardo).

En concreto, aquí se discute el planteamiento del problema, el modelo Simulink y su evaluación en bucle abierto y con un regulador inicial subóptimo. La optimización y análisis de la respuesta temporal son discutidas en el vídeo [591].

La primera parte del vídeo discute el comando `sim` de Matlab, que simula modelos de Simulink (ficheros con extensión `.slx`); si alguno de los bloques tiene parámetros que dependen de variables, deben estar definidas en el *Base Workspace* de Matlab; cuando se hacen simulaciones dentro de una función (que tiene su propio Workspace), los parámetros de los modelos deben asignarse mediante el comando `assignin`.

La primera ejecución tarda bastante, debido a cargas de librerías y compilaciones. Las siguientes tardan menos, y tardan todavía menos si se activa la opción `FastRestart`.

A continuación, se discuten los detalles de la configuración de los diferentes elementos del diagrama de bloques, tanto del escenario sobre el que se desea simular (rechazo de perturbación tipo escalón a la entrada del proceso), como del cálculo de los índices de

coste arriba mencionados. El controlador elegido es un PI, $C(s) = K_c \cdot (1 + \frac{1}{T_i s})$, siendo K_c y T_i los parámetros ajustables (variables de decisión) que se desean calcular para optimizar prestaciones.

Por último, se simula y calculan los costes en bucle abierto ($K_c = 0$), y también con un controlador inicial ($K_c = 1.6, T_i = 35$), para comprobar que el código que prepara el problema para los optimizadores funciona correctamente, y para hacerse una idea del funcionamiento del proceso.

[591: opt4idxb] Optimización ISE/IAE/ITAE/IT2AE para rechazo de perturbaciones: comparativa de respuesta temporal

X. Blasco, J.M. Herrero, C. Ramos, A. Sala (UPV)

Materiales: [[Cód.:](#) EjemploPI4IndicesFR.zip] [[PDF](#)]

*** 10:59

[PLAY ►]

En este vídeo se continua el caso de estudio sobre la comparación de la respuesta temporal resultante de optimizar cuatro índices diferentes (ISE, IAE, ITAE, IT2AE) para un mismo proceso (1er orden + retardo). El detalle sobre el planteamiento del problema (diagrama de bloques) y comandos `sim`, `assignin` se aborda en el vídeo [590]; no obstante, un lector familiarizado con Simulink podría no necesitar visionarlo y tener suficiente con el resumen rápido del problema que se hace en los dos primeros minutos de este vídeo.

Tras esa revisión, se aborda la optimización del ISE con un optimizador local (`fminunc`, Optimization Toolbox), comenzando en el regulador inicial en el que se había simulado en las primeras pruebas. También se optimiza con el comando `ga` de la Global Optimization Toolbox. Se comprueba que convergen al mismo mínimo, con lo cual existen altas probabilidades de que sea el mínimo global; el `ga` tarda 30 veces más que el `fminunc` con las opciones y tolerancias usadas en este caso particular (la idea es la misma que en el ejemplo introductorio del vídeo [927]).

A partir del minuto 4, se optimiza con `fminunc` los otros índices (IAE, ITAE, IT2AE) usando como semilla inicial el valor de las constantes que daban el ISE óptimo. Se comparan los índices con el bucle abierto y con el regulador inicial, y se simula la respuesta temporal de la salida de todas las soluciones.

Se observa que como elevar al cuadrado aumenta la importancia de los errores “grandes”, el óptimo ISE es el que mejor (más bajo) tiene el pico (valor máximo) del error. No obstante, el PI resultante tiene unas ganancias más altas que el resto que producen un transitorio menos amortiguado (más oscilatorio) que las otras opciones. Debido a la ponderación temporal, el IT2AE consigue el tiempo de establecimiento más rápido, como era de esperar intuitivamente.

Se comenta que, dado que la variable manipulada u no aparece en el índice de coste, el controlador óptimo podría tener una respuesta de perturbación a control inadecuada. En este caso, se analiza dicha respuesta temporal y se considera razonable; si no lo fuera, podría incluirse algún término en el índice de coste relacionado con la acción de control.

La conclusión es que la elección del índice a optimizar es el paso clave en control óptimo: reducir todos los requerimientos de error pequeño, amortiguamiento, amplificación de ruidos de medida, robustez, etc. a un único número parecía fácil, pero no lo es tanto en todas las aplicaciones.

16.2.2 Control óptimo via ode45+fminunc (ejemplo Matlab)

[592: optode1] Optimización de parámetros de reguladores con ode45+fminunc: planteamiento del problema y ejemplo

Materiales: [[Cód.:](#) optimconode.mlx] [[PDF](#)]

*** 10:10

[YouTube ►]

Dado que Simulink internamente transforma los diagramas de bloques a código usable por los integradores numéricos (`ode45`, etc.), si dichas manipulaciones se realizan por el programador, se pueden resolver problemas de control óptimo usando el mencionado `ode45` y el comando `fminunc` de la Optimization Toolbox (o, opcionalmente, `fmincon` si se desea incluir restricciones o cotas que determinen la región de parámetros a explorar).

Si el lector no tiene familiaridad con la representación interna (variables de estado) de sistemas dinámicos e integración numérica, se le recomienda visionar los vídeos [10], [16] y [32] antes de continuar con éste.

Este vídeo plantea un problema de control óptimo sobre un modelo de péndulo en su punto de equilibrio superior inestable. Como `ode45` trabaja con la representación en variables de estado $\frac{dx}{dt} = f(t, x)$, se detallan:

- El modelo en representación interna en bucle abierto $\frac{dx}{dt} = f(x, u)$ invariante en el tiempo, siendo el vector de estado el vector $x = (p, v)^T$ con la posición y la velocidad.
- La parametrización elegida (regulador PD, pero como la referencia es constante igual a cero –pto. func.– se puede simplificar a $u = -K_p p - K_v v$, que en teoría de control se denomina *realimentación del estado*. El vector $\theta = (K_p, K_v)$ serán los parámetros ajustables de este problema, expresando el controlador con $u(x, \theta)$.
- Las ecuaciones en bucle cerrado: $\frac{dx}{dt} = f(x, u(x, \theta)) = g(x, \theta)$.
- El índice de coste cuadrático $c(x, u) = q_1 p^2 + q_2 v^2 + r u^2 = q_1 x_1^2 + q_2 x_2^2 + r u^2$ a utilizar (objetivo: minimizar su integral)... en la literatura, se suele denominar a $c(x, u)$ el coste inmediato, para distinguirlo del coste acumulado $J = \int_0^{T_f} c(x(t), u(t)) dt$.

Este vídeo discute las fases iniciales de modelado y planteamiento del problema; el código de simulación y optimización es discutido en detalle en el vídeo [593], continuación de éste.

Nota: En este vídeo, el controlador no tiene dinámica (es “realimentación del estado”, posición y velocidad medidas ambas); el vídeo [355] simula un bucle cerrado con un regulador dinámico construyendo las ecuaciones de estado en bucle cerrado con código Matlab, sin interfaz gráfica de usuario, a partir de las ecuaciones de estado del proceso y el controlador. Se aconseja su visualización para comprender cómo se optimizarían controladores con estado interno.

[593: optode2] [Optimización de parámetros de reguladores con ode45+fminunc](#)

(2): Simulación y optimización

Materiales: [[Cód.: optimconode.mlx](#)] [[PDF](#)]

**** 11:00

[YouTube ▶]

Este vídeo es continuación del [592], donde se modelaba un péndulo invertido, se proponía una expresión paramétrica de un controlador $u(x, \theta)$ y se planteaba un índice de coste “inmediato” $c(x, u) \geq 0$ cuya integral J se deseaba minimizar.

En este vídeo se plantean los detalles para solucionar el problema con `ode45` y `fminunc`. En concreto, se extiende el estado añadiendo el coste acumulado al vector de estado como una tercera variable de estado, debido a la existencia de la nueva ecuación diferencial $\frac{dJ}{dt} = c(x, u)$, que debe ser añadida al modelo. Se presenta el código que realiza dicha modificación al modelo, la integración numérica y la presentación gráfica de resultados.

El objetivo de la optimización será minimizar el valor de J al final del tiempo de simulación. Para ello, se genera una función $J(\theta)$ a la que le entran dos números $\theta := (K_p, K_v)$, ejecuta la simulación y devuelve el valor final del tercer estado (coste acumulado) del integrador numérico (este “estado” es una variable sin significado “físico”, obviamente). Con esa función $J(\theta)$, una llamada a `fminunc(J, \theta)` obtiene el valor óptimo de los parámetros que minimizan el objetivo buscado (aunque es posible que sea un mínimo local en las proximidades del punto de exploración inicial elegido por el usuario).

La parte final del vídeo compara la respuesta temporal en bucle cerrado del regulador óptimo con la obtenida con los parámetros iniciales escogidos al azar para comenzar la búsqueda.

Es bien conocido que la optimización está detrás de muchos problemas de ingeniería. En particular, la optimización de parámetros de modelos para ajustar datos experimentales (que recibe el nombre *identificación experimental*) aplica Ideas muy similares a las aquí discutidas; se recomienda visualizar los vídeos [749] y [750] para ver usos de `ode45+fminunc` o, con restricciones en la región de búsqueda de los parámetros `ode45+fmincon`, en identificación (bueno, de hecho, esos vídeos plantean el uso de un algoritmo genético de búsqueda global aleatoria dirigida que también podría haberse usado aquí).

Nota: En este vídeo, el controlador no tiene dinámica; el vídeo [355] simula un bucle cerrado con un regulador dinámico construyendo las ecuaciones de estado en bucle cerrado con código Matlab, sin interfaz gráfica de usuario, a partir de las ecuaciones de estado del proceso y el controlador.

También cabe indicar que la eficiencia computacional de la integración numérica es importante en optimizaciones complejas; en algunos casos (stiff) de sistemas de orden alto con constantes de tiempo muy diferentes, `ode15s` podría funcionar mejor que `ode45`, ver vídeo [39].

[594: optode3] Optimización de parámetros de reguladores con `ode45+fminunc`
(3): comparación con solución LQR, modificación índice de coste

Materiales: [[Cód.: optimconode.mlx](#)] [[PDF](#)]

**** 09:00

[YouTube ►]

Este vídeo es el tercero y último de la serie que incluye a los vídeos [592] y [593] sobre el caso de estudio de control óptimo minimizando la integral de un índice cuadrático sobre un péndulo invertido no lineal mediante `ode45` y `fminunc`.

La solución de este problema en el caso lineal es el celebrado regulador óptimo **LQR** (Linear Quadratic Regulator), cuyos detalles teóricos no entran en los objetivos de este ejemplo Matlab. Dicha solución se obtiene utilizando el comando `lqr` de la Control Systems Toolbox. La primera parte del vídeo compara los parámetros óptimos de `fmincon` y los de `lqr` viendo que son similares (coincidirían para condiciones iniciales muy cercanas al origen y horizonte de simulación suficientemente largo). La solución LQR es, en general, más eficiente computacionalmente que la de `fmincon` pero sólo aplica a procesos lineales y coste cuadrático, mientras que `fmincon` es genérico para cualquier modelo o costes no lineales (con ciertas suposiciones mínimas de suavidad).

En la segunda parte del vídeo, se presenta el efecto de modificar el índice de coste para traducir el significado “formal” de optimalidad a un significado “práctico” de rapidez *vs.* limitaciones en velocidades y fuerzas máximas durante el transitorio. Disminuyendo la ponderación en el índice de velocidades y fuerzas se consiguen bucles más rápidos, como se esperarí intuitivamente.

Control óptimo de procesos inciertos: El control robusto trata de diseñar reguladores que se comporten bien ante errores de modelado. Una opción es optimizar no sólo el resultado de una simulación (como se ha hecho en esta sección), sino simular el mismo regulador con *varios* procesos diferentes y optimizar un índice que dependa de las prestaciones obtenidas en ese conjunto de simulaciones... usualmente se quiere que la *peor* de dichas simulaciones tenga, aún así, un comportamiento aceptable. Las ideas de esta sección son extendidas al caso robusto en la sección 25.3.

Nota: Aparte de “Simulink Response Optimization” o de “`ode45+fminunc`” está la opción de usar “`fmincon`” para acotar el espacio de búsqueda de parámetros, o el de simular un modelo de simulink desde la línea de comandos con el comando `sim`, como una opción adicional para poder controlar la optimización desde un fichero de código sin necesidad de generar el modelo en formato `ode45`. Se deja al lector los detalles, aunque también puede consultar los vídeos [optrobpi1(10:42)] y [optrobpi2(08:41)] donde se sigue exactamente este camino (aunque se plantean ideas de optimización robusta fuera de los objetivos de esta sección).

16.3 Control multivariable LQR

El control óptimo LQR (que luego dio lugar al control predictivo), es una estrategia de control clave. Está bien desarrollada en muchos materiales, y no tengo material propio preparado.

Asimismo, un enfoque moderno al control óptimo lineal (que generaliza LQR, Kalman,...) es el control óptimo \mathcal{H}_2 y \mathcal{H}_∞ , discutido en el Capítulo 19 de estos materiales.; cabe recalcar que el control óptimo \mathcal{H}_∞ es la base de los desarrollos de control robusto (garantizar tolerancia a errores de modelado) que se desarrolla en capítulos posteriores.

El único material propio del que, por el momento, dispongo es el relativo a la solución del problema basada en desigualdades matriciales lineales (ver Capítulo 31), a continuación:

[595: lqrlmi]

Control óptimo (LQR) de procesos lineales con desigualdades matriciales lineales (LMI): teoría y ejemplo Matlab

Materiales: [[CÓD.: lqrlmitest.mlx](#)] [[PDF](#)]

**** 09:35

[YouTube ▶]

Este vídeo revisa la teoría que permite resolver el problema LQR en tiempo continuo (optimizar $\int_0^\infty x^T Q x + u^T R u dt$ con un cierto $u(x)$, en particular $u = -Kx$) mediante desigualdades matriciales lineales, a partir de la desigualdad escalar $\frac{dV}{dt} + x^T Q x + u^T R u \leq 0$, en un proceso lineal $\dot{x} = Ax + Bu$.

Una vez revisada la teoría, se hace un ejemplo numérico sobre un proceso de orden 4, para comprobar que la solución coincide con el comando `lqr` de Matlab. Se utilizan `Yalmip` y `SeDuMi` para los cálculos (ver <https://www.tbxmanager.com/> para instalación).

La teoría de este vídeo puede generalizarse muy fácilmente a sistemas lineales con parámetros inciertos, obteniendo controladores (*sub*)óptimos denominados control de *coste garantizado* que garantizan prestaciones robustas, ver vídeo [905]; también se puede generalizar a algunos casos no-lineales o variantes en el tiempo mediante lo que se denomina *planificación de ganancia*, ver vídeos [906] y [909].

16.4 Control Predictivo (enlaces a terceros)

El control predictivo, del que no tengo material propio preparado, se ha convertido en una **estrategia de control muy importante** en aplicaciones prácticas que, obviamente, debe ser estudiada por el lector en otros materiales. Por ejemplo, en el enlace a continuación hay un curso completo de 20 vídeos, con más de 12 horas de exposición sobre **control predictivo basado en modelo (MPC)**: <https://www.youtube.com/playlist?list=PLF-qcfymUY4U8heDB4TqzPUHUs-PaN2EM>

Nota: La *identificación* de modelos dinámicos multivariantes lineales expresamente orientados al control predictivo sí se aborda en mis materiales, en concreto en los vídeos [idmpc1(11:09)] y [idmpc2(10:52)].

16.5 Control óptimo no lineal (enlaces a terceros)

En este otro enlace podéis seguir una serie de lecciones (14 videos, aprox. 2.5 horas de contenido) sobre **Control Óptimo, Optimización Dinámica**:


<https://www.youtube.com/watch?v=Tk03YXaemFo&list=PLXmyaEzsglCQp0wVTxfiFuLdUqR7K0GiP>

Este curso está enfocado al caso no lineal continuo (Hamiltoniano, cálculo de variaciones, etc.).

16.6 Programación dinámica (En elaboración)

[596: dp2d1] Ejemplo Matlab Programación Dinámica (1): laberinto 2D, modelado y planteamiento del problema.

Materiales: [[CÓD.:](#) DP2dTests.mlx] [[PDF](#)]

***  07:04

[YouTube ►]


Este video plantea un problema de programación dinámica sobre un modelo de 24 estados (laberinto 2D de 6x4 celdas) y cuatro acciones de control (arriba, abajo, izquierda, derecha).

Se plantea el código Matlab del modelo, el coste inmediato, el factor de descuento ($\gamma < 1$) y el objetivo de coste a minimizar, y se construye una tabla de transiciones (estado origen)–(acción)–(estado destino), que será usada por los algoritmos posteriores.

La resolución del problema con programación dinámica se aborda en el vídeo [597], continuación de éste.

[597: dp2d2] Ejemplo Matlab Programación Dinámica (2): laberinto 2D, iteración de valor, programación lineal.

Materiales: [[CÓD.:](#) DP2dTests.mlx] [[PDF](#)]

***  13:29

[YouTube ►]


Este video resuelve un problema de programación dinámica sobre un modelo de 24 estados (laberinto 2D de 6x4 celdas) y cuatro acciones de control (arriba, abajo, izquierda, derecha). El detalle del modelado y planteamiento del problema aparece en el vídeo [596], cuya visualización previa podría ser aconsejable.

En este vídeo se presenta código para resolver de forma exacta el problema (calcular la función de valor óptima en los 24 estados) por:

- Iteración de Valor (Value Iteration). Primero se esboza con "tres bucles anidados" pero el código final se presenta vectorizado (paralelizado) para mayor eficiencia.
- Programación Lineal. Se observa que el tiempo de cómputo es mayor, al menos con la implementación y versión de Matlab usadas aquí.

[598: dp2d3] Ejemplo Matlab Programación Dinámica (3): laberinto 2D, iteración de valor (policy iteration)

Materiales: [[CÓD.:](#) DP2dTests.mlx] [[PDF](#)]

****  13:27

[YouTube ►]

Este vídeo resuelve el problema planteado en el vídeo [596] mediante iteración de política (policy iteration), complementando a la iteración de valor (value iteration) y la programación lineal discutidas en el vídeo [597].

En la fase de "evaluación" de una política se plantean dos opciones: a) hacerlo resolviendo un sistema de ecuaciones o b) mediante iteración de valor.

Todas las opciones producen el mismo resultado para política y función de valor óptima (tolerancias aparte) que los otros algoritmos.

Capítulo 17

Predicción óptima en procesos con perturbaciones aleatorias (filtro de Kalman)

17.1 Preliminares

El concepto de “observador”, introducido en el vídeo [obs(13:55)], cuando las entradas son ruido blanco dio lugar a lo que se denomina Filtro de *Kalman*, abordado en esta sección¹. Si el concepto de *observador del estado* no le es familiar al lector, se recomienda visionar el vídeo [obs(13:55)] que motiva el problema que se resuelve a continuación.

[599: obsruido] Observadores del estado: análisis de efecto de ruidos de proceso y medida *** 08:57
Materiales: [EfectoRuidoEnObservadores.pdf] [YouTube ▶]

En este vídeo se discute el efecto del ruido de proceso w de un sistema “real” dado por $x_{k+1} = Ax_k + Bu_k + w_k$ y del ruido de medida v de sensores $y_k = Cx_k + v_k$, sobre los estimados del estado que calcula un observador discreto adelantado $\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_{k+1} - C(A\hat{x}_k + Bu_k))$; las ecuaciones de este observador han sido introducidas en el vídeo [561].

Existen tres objetivos de diseño, en algunos casos contradictorios: (1) Anular con rapidez el efecto de condiciones iniciales de error no nulas; (2) Usar los sensores para anular el efecto del ruido de proceso; (3) Usar el modelo para filtrar, promediando durante un cierto tiempo, el ruido de medida.


Estos tres objetivos son discutidos analizando la ecuación del error resultante: $e_{k+1} = (A - LCA)e_k - (I - LC)w_k + LCv_{k+1}$.

El filtro óptimo que minimiza la varianza estacionaria (cuando el efecto de las condiciones iniciales ha desaparecido) es el *Filtro de Kalman*, cuyo desarrollo teórico se aborda en el vídeo [604].

¹Existen alternativas al filtro de Kalman para el problema de estimación del estado: la solución por asignación de polos, en un entorno determinista, se discute en el vídeo [obsap(20:29)], la solución con una interpretación frecuencial \mathcal{H}_∞ se aborda en el vídeo [hinfobs1(09:41)]. Cada una de ellas parte de unas suposiciones a priori y objetivos diferentes sobre el error (diferencia entre estado real y estimado).

17.2 El filtro de Bayes recursivo (motivación y ejemplo ‘tigre oculto’)

Como ejemplo motivador de la idea sobre un problema elemental, en esta sección se llegará a la fórmula más sencilla de filtro recursivo bayesiano recursivo a partir del caso de un ‘tigre oculto’ que ya sirvió para motivar cuestiones relacionadas con probabilidad conjunta, condicional, marginal, etc. en la revisión de conceptos básicos de estadística, ver vídeos [tiger1(20:57)], [tiger2(15:52)] y [tiger3(21:57)].

[600: tiger4]	Tigre oculto (4): Probabilidad condicional y fórmula de Bayes DOS rugidos	**  14:19
Materiales: [Cód.: BayesNORecursivoParte2.mlx] [PDF]		[YouTube ►]

*[LINK TO ENGLISH VERSION](#)


Este vídeo es parte del caso de estudio “tigre” que comenzó en el vídeo [947], que continua la inferencia mediante fórmula de Bayes discutida en el vídeo [952], el inmediatamente anterior a este.

En este vídeo se aborda el problema de estimar la probabilidad a posteriori de localización de un “tigre oculto” tras escuchar DOS rugidos (sin permitir que el tigre se mueva de lado entre rugidos), mientras que en vídeos anteriores sólo se suponía que se escuchaba UN rugido.

El problema se abordará formando la tabla de probabilidad condicional de las cuatro posibles observaciones (LL, LR, RL, RR) condicionadas a la posición en la jaula izquierda o derecha del tigre.

Para formar dicha tabla haremos la hipótesis de “independencia condicional” que permitirá multiplicar probabilidades condicionales; se discute brevemente su significado en el contexto del problema del tigre (si se da la condición de que el tigre no cambia de lado, entonces sucesivos rugidos son independientes), pero se remite a los vídeos [961] y [962] para un análisis en detalle del concepto.

Una vez esta nueva tabla condicional se ha formado, la aplicación de la fórmula de Bayes es idéntica a casos anteriores. El vídeo [601] discutirá el caso general de un número de rugidos arbitrarios, y el vídeo [602] propondrá una fórmula de Bayes recursiva para resolver el mismo problema.

[601: tiger5]	Tigre oculto (5): fórmula Bayes N rugidos (N arbitrario) NO recursiva	***  17:46
Materiales: [Cód.: BayesNORecursivoParte2.mlx] [PDF]		[YouTube ►]

*[LINK TO ENGLISH VERSION](#)

Este vídeo es parte del caso de estudio “tigre oculto” de la colección. Aquí se generaliza la fórmula de Bayes que se vio para el caso de un rugido en el vídeo [952] y para dos rugidos en [600].

En concreto, se generaliza a un número arbitrario de rugidos. Se introducen dos posibles tablas de probabilidad condicional: una expandida donde el orden en el que los rugidos se reciben sí importa, y otra comprimida donde sólo importa el número de rugidos a izquierda o derecha escuchados; la segunda tabla tiene distribución “binomial”.

La aplicación de la fórmula de Bayes en ambos casos produce un resultado idéntico. El vídeo finaliza con un ejemplo numérico que ilustra que si esperamos a escuchar unos 15 rugidos entonces tenemos prácticamente certeza total sobre en qué lado está el tigre (suponiendo que no puede cambiar de jaula entre rugidos).

[602: tiger6br]	Tigre oculto (6): fórmula Bayes Recursiva (demostración)	****  17:31
Materiales: [Cód.: BayesRecursivo2.mlx] [PDF]		[YouTube ►]

*[LINK TO ENGLISH VERSION](#)

Este vídeo revisa la teoría que justifica la fórmula de Bayes recursiva para el problema de N rugidos de un tigre oculto. La fórmula NO recursiva se discutió en el vídeo [601]. Aquí se comprueba que aplicando N veces la fórmula de Bayes con la tabla condicional de ‘1 rugido’ se obtiene la misma probabilidad a posteriori final que aplicando una única vez la tabla de ‘ N rugidos’. En ambos casos, se debe suponer ‘independencia condicional’ para que los resultados sean correctos.

[603: tiger7rbs] Tigre oculto (7): fórmula Bayes Recursiva (simulación Matlab) *** 17:44
Materiales: [Cód.: BayesRecursivov2.mlx] [PDF] [YouTube ►]

*LINK TO ENGLISH VERSION

Este vídeo presenta una simulación del filtro de Bayes recursivo en el problema del tigre oculto que estamos analizando en varios vídeos precedentes. En concreto, la teoría del filtro recursivo se discutió en el vídeo [602]; bueno, solamente una versión simplificada adaptada al caso de estudio en consideración.

Aquí se detalla el código para llevarla a cabo y se discuten sus resultados, obviamente coincidentes con los de la fórmula no recursiva del vídeo [601].

Se hacen varias simulaciones con diferente relación “señal-ruido” de las observaciones (porcentaje de rugidos que se escuchan por el lado equivocado). Cuanto más ruidosas son las observaciones, más rugidos se necesitan para estar seguros de dónde se encuentra escondido el tigre. En todas las simulaciones, el tigre permanece siempre en el mismo lado, dado que así se necesita que sea según las suposiciones para llevar a cabo la inferencia estadística. La versión donde el tigre cambia de lado se verá en futuros materiales.

17.3 Predicción óptima en procesos gaussianos en tiempo discreto

Nota (prerrequisitos): Para comprender en su totalidad los desarrollos teóricos del filtro de Kalman a continuación, es muy recomendable el revisar los conceptos básicos sobre procesos estocásticos lineales discretos (Sección 7.2) y sobre predicción estadística, discutidos en la Sección B.5.

[604: kalteo] Predicción óptima en sistemas dinámicos lineales: Filtro de Kalman tiempo discreto (demostración) *** 15:46
Materiales: [FiltroKalman.pdf] [YouTube ►]

En este vídeo se revisan las fórmulas de mejor predicción lineal de una variable dada otra, en base a la matriz de varianzas-covarianzas conjunta (vídeo [967]), que coinciden con mínimos cuadrados en un enfoque determinista (vídeo [971]) que aquí no será utilizado. Luego, se aplican para obtener la mejor predicción del estado x de un sistema dinámico dado un conjunto de medidas y (el sistema tiene ruido de proceso y ruido de medida). La simulación en bucle abierto de dichos procesos se presentó en el vídeo [219]; este vídeo reescribe las fórmulas, generalizándolas para incorporar el efecto de la información proporcionada por sensores sobre la predicción del estado interno del proceso.


Nota: aunque no se dice explícitamente en el vídeo, se utiliza la suposición de *independencia condicional* de las medidas y el siguiente estado entre sí y con respecto a cualquier otra variable pasada, según se discute de forma genérica en el vídeo [962]. Ello implica que toda la información del pasado puede “condensarse” en la variable intermedia x_k para predecir el futuro (se ha añadido una transparencia al PDF).

En concreto, a partir de un estimado \hat{x}_{k-1} , $\Sigma_{x_{k-1},p}$ (condiciones iniciales) suponiendo que incorpora toda la información conocida de medidas hasta el instante $k-1$, se calcula la

ecuación de medias del observador óptimo que produce \hat{x}_k en función de dichas condiciones iniciales y la medida y_k , y también se actualiza la matriz de varianzas-covarianzas incorporando la información del sensor, obteniendo $\Sigma_{x_k,p}$. Con ello, se puede aplicar recursivamente utilizando y_{k+1}, y_{k+2}, \dots y dicho algoritmo constituye el *observador óptimo* o *Filtro de Kalman*, muy utilizado en gran cantidad de aplicaciones de control y monitorización, siendo uno de los resultados más importantes de la estadística multivariable lineal.


Nota 1: El filtro de Kalman estacionario es un caso particular de lo que se denomina problema \mathcal{H}_2 de control óptimo, como se discute en el vídeo [668].

Nota 2: La generalización del filtro de Kalman a sistemas no lineales, mediante linealización alrededor de trayectorias (vídeo [875]), se denomina filtro de Kalman extendido (detalles en vídeo [616]).

Nota: El filtro de Kalman puede ser usado para estimar la media de una variable. Por ejemplo, planteando el modelo $m_{k+1} = m_k$ (ec. de estado), $y_k = m_k + v_k$ (ec. de salida) se está diciendo que la media de y es m_k y se mantiene constante, y a partir de un estimado inicial *a priori* puede refinarse con futuras observaciones. Este enfoque, en el contexto de “test estadístico de media” en detección de fallos, se desarrolla en el vídeo [thmdkal(15:25)], cuya visualización, si se dispone de tiempo, se aconseja en este momento. 

A continuación se presentan unos primeros ejemplos Matlab de uso del filtro de Kalman; la Sección 17.4 presenta ejemplos adicionales.

[605: tubokal]
Filtro de Kalman estacionario: ejemplo estimación temperatura de gas en tubería (Matlab)

 12:43

Materiales:
[CÓD.: ModeloTuboKalman.mlx] [PDF]
[YouTube ►]

Este vídeo explora la salida del comando dlqe de matlab que calcula las matrices de ganancia de un filtro de Kalman (estacionario) para un proceso discreto.


El proceso de ejemplo es el sistema térmico (tubería) modelado en el vídeo [273]; las características estadísticas (varianzas) de la respuesta se discutieron en el vídeo [220]. Los dos primeros minutos de este vídeo resumen rápidamente las conclusiones principales de los dos vídeos referenciados.

Se diseñan tres opciones diferentes para los filtros de Kalman:

1. Con un sensor “inservible” con varianza de ruido de medida tendiendo a ∞ : el resultado es una ganancia cero del observador y unos estimados del estado idénticos al bucle abierto calculado en el vídeo [220].
2. Sensor “perfecto” con varianza de ruido de medida prácticamente cero: el resultado es una ganancia elevadísima del observador, y unos polos de bucle cerrado rápidos que dan lugar a unas varianzas estimadas de los estados internos muy pequeñas.
3. Las dos situaciones anteriores son “extremas” y en el tercer diseño se calculan las propiedades estadísticas de un filtro de Kalman “realista” con un ruido de medida en la sonda con desviación típica de 0.6 grados. También se simula el observador resultante (construido con el comando kalman) y se comprueba la validez del intervalo de confianza 95% del error en la estimación de la temperatura del gas que circula por la tubería (2 veces la desviación típica).

Nota: Cuando no se cumplen las suposiciones teóricas subyacentes en la metodología, el filtro de Kalman puede no funcionar todo lo bien que se esperaría, ver vídeo [614] para un ejemplo en dicho sentido.

[606: kalml]
Filtro de Kalman: ejemplo Matlab (filtro no estacionario)

 33:33

Materiales:
[CÓD.: kalmtst2masas.mlx] [PDF]
[YouTube ►]

Este vídeo presenta cómo simular el filtro de Kalman no estacionario discutido en el vídeo [604], sobre un proceso de masas-muelles lineal ya considerado, por ejemplo, en el vídeo [221].

En concreto, dicho vídeo discutía la simulación en bucle abierto, sin mediciones. La incorporación de sensores hace que, obviamente, la matriz de varianzas-covarianzas del error de estimación sea mucho más pequeña que la que resultaba en el vídeo [221]; los resultados coincidirían si la matriz de varianzas-covarianzas del “ruido de medida” tendiera a infinito (sensores “inútiles”).

El modelado y revisión de conceptos de teoría abarca hasta el minuto [03:55].

Aunque el vídeo discute la simulación de un filtro *no estacionario*, se comenta en el mismo que en muchas aplicaciones se implementa el filtro *estacionario* (ganancias constantes, observador invariante en el tiempo) y, por tanto, el objetivo del vídeo es una implementación “purista” de los desarrollos teóricos para comprobar su corrección, y no la implementación estacionaria de ganancia constante en aplicaciones, que utilizaría el código de los observadores del vídeo [562] cambiando la línea `place` por `dlqe`.

En efecto, el código compara la solución obtenida (excepto en un transitorio de dos segundos) con la del comando `dlqe` y son coincidentes.

La parte final del vídeo compara la calidad de los estimados según la configuración de sensores disponible: posición 1 *vs.* posición 2 *vs.* ambas. Las desviaciones típicas (raíz cuadrada de la diagonal de la matriz de varianzas-covarianzas del error de observación) nos indican cómo de útil es cada configuración de sensores para poder estimar cada estado, según la precisión conseguida.

Nota 1: El filtro de Kalman está *muy* relacionado con los mínimos cuadrados *recursivos* en identificación de θ dado un modelo $y = x^T \theta$, a partir de series de datos (y_1, y_2, \dots) , (x_1, x_2, \dots) . En efecto, el filtro de Kalman es el estimador óptimo del proceso lineal $\theta_{k+1} = \theta_k$ (suponer parámetros constantes, esta vez entendido como una ecuación de *estado*), $y = x^T \theta$ (entendido como una ecuación de salida donde el “estado” es θ). Los vídeos [mcr1(11:18)] y [mcr2(10:51)] discuten dicha relación en detalle. Suponiendo un hipotético ruido de proceso $\theta_{k+1} = \theta_k + w_k$ ello permite incorporar un “olvido” de la información pasada (discutido en el vídeo [mcr2(10:51)]).



Nota 2: El observador óptimo estacionario (filtro de Kalman `dlqe`) y el controlador óptimo de horizonte infinito `dlqr` están muy relacionados con lo que se conoce en la literatura como control óptimo \mathcal{H}_2 . Específicamente sobre la parte de observación (observador \mathcal{H}_2), el vídeo [h2obsml(11:31)] discute (con un ejemplo de código) la equivalencia entre cierto problema de optimización \mathcal{H}_2 y el filtro de Kalman estacionario. Si cerramos el bucle (principio de separación), La equivalencia entre “Kalman+`lqr`” (estacionario) y control \mathcal{H}_2 se ilustra también mediante un breve ejemplo Matlab en el vídeo [h2lqrml(27:21)].

17.3.1 Predicción óptima no causal: suavizado RTS

[607: rtsm] Suavizado no causal en representación interna (RTS smoother)

**** 14:05

Materiales: [RTSsmooth.pdf]

[YouTube ▶]

Este vídeo presenta el problema del suavizado no causal en representación interna y la solución Rauch-Tung-Striebel (RTS smoother). El suavizado RTS calcula la mejor predicción del estado basada en información de medidas pasadas (como Kalman) y *futuras*, si están disponibles. Este suavizador generaliza las ideas de filtros no causales en función de transferencia (ver comando `filtfilt` de Matlab, por ejemplo, o vídeo [267]).

Se desarrolla la demostración teórica de las fórmulas RTS, que acaban siendo una iteración hacia adelante en el tiempo del filtro de Kalman causal junto con una iteración hacia atrás en el tiempo de un paso de suavizado/corrección posterior. Aunque no se mencionan explícitamente en el audio, determinadas suposiciones de independencia condicional (modelo de Markov, espacio de estado) son obviamente necesarias.

17.4 Casos de estudio Kalman/RTS

17.4.1 Caso 1: estimación de fuerza aplicada sobre un sistema masa-muelle

[608: estfml1] Estimación de fuerza de entrada a un sistema mecánico (1):
planteamiento del problema y filtro de Kalman estacionario

Materiales: [[CÓD.:](#) KalvsRTSsmootherEx1b.mlx] [[PDF](#)]

*** 10:32

[YouTube ▶]

Este vídeo plantea un sistema masa-muelle-amortiguador modelado como $G(s) = 2/(s^2 + s + 5)$, y discretizado a $T_s = 0.04$ s con retenedor de orden cero. Existe una fuerza de entrada conocida u , y otra fuerza desconocida w (ruido de proceso) que se desea estimar, dado un sensor con ruido de medida.

El vídeo plantea la inutilidad de estimar la fuerza resultante mediante la segunda derivada numérica de la medida, debido al ruido. Por ello, plantea un modelo de orden 3 para sintetizar un filtro de Kalman estacionario. En concreto, el polo adicional se introduce porque se hace la suposición de que la señal w tiene un ancho de banda de 0.4 rad/s... realmente no es así (es un tren de escalones), pero el filtro de Kalman necesita suponer que las perturbaciones son ruido con distribución normal.

Se usa el comando `kalman` para sintetizar el filtro estacionario y se analiza su respuesta en frecuencia. El análisis de respuesta temporal se aborda en el vídeo [609] continuación de este.

[609: estfml2] Estimación de fuerza de entrada a sistema mecánico (2):
Kalman no estacionario y suavizado RTS

Materiales: [[CÓD.:](#) KalvsRTSsmootherEx1b.mlx] [[PDF](#)]

**** 11:00

[YouTube ▶]

Este vídeo es continuación del [608]. En este se simula el filtro de Kalman no estacionario (en poco tiempo las ganancias del observador convergen al estacionario). Se comprueba el ajuste de la estimación de posición, velocidad y fuerza de entrada w en función del ancho de banda supuesto de dicha perturbación w .

En la segunda parte del vídeo se usa un suavizado Rauch-Tung-Striebel para filtrar mejor las señales con información no causal y tener un mejor estimado de la fuerza w .

Nota: El vídeo [614] muestra un ejemplo donde el filtro de Kalman funciona muy mal (peor que una derivación numérica filtrada “a ojo”) sobre un proceso de segundo orden similar al aquí estudiado, si no se cumplen las suposiciones teóricas subyacentes.

17.4.2 Caso 2: eliminación de tendencias (rampa)

Esta sección usa un modelo de generador de perturbaciones ([ress(10:57)]) para eliminar una “tendencia” en forma de rampa de una señal, comparando diferentes enfoques. La comparación con la asignación de polos (vídeo [drngenml(14:08)]) se deja al lector como ejercicio propuesto.

[610: rampdetr] Eliminación de derivas y rampas (1): detrend

Materiales: [[CÓD.:](#) ramp1detrend.zip] [[PDF](#)]

*** 07:57

[YouTube ▶]


Es este vídeo se plantea el problema de eliminar derivas constantes o rampas (deriva en velocidad) de una serie de datos . Se presenta un modelo de doble integrador que genera ese tipo de perturbaciones (aceleración ruido blanco, doble integrador) y se genera una serie temporal ejemplo de la cual esa tendencia debe ser eliminada.

La primera solución que se planteará será el estimar una recta de regresión por mínimos cuadrados y restarla a la señal. Esto lo hace también el comando `detrend` . También se utiliza dicho comando para ajustar un polinomio "medio" de grado 3 para intentar eliminar la tendencia observada en los datos.

En los vídeos [611] y [612] se abordará el mismo problema por medio del filtro de Kalman.

[611: rampkales]

Eliminación de derivas y rampas (2): filtro de Kalman
 estacionario

****  12:38

Materiales: [[CÓD.:](#) ramp2KalmanEstacionariodlqe.mlx] [[PDF](#)]


[\[YouTube ►\]](#)

Este vídeo aborda el mismo problema que el vídeo [610], mediante un filtro de Kalman estacionario, con el comando de Matlab `dlqe` . Se utiliza el modelo de doble integrador que en el referido vídeo se usaba para generar la deriva/tendencia, esta vez incorporado a las ecuaciones de estado del modelo para diseñar un observador óptimo, de modo similar a la propuesta en los vídeos [165] y [166] con asignación de polos.

El resultado es un filtro paso-alto (doble derivador, $(z-1)^2$), que concuerda con lo intuitivamente esperado: el doble derivador cancela el efecto del doble integrador. Se comprueba la respuesta en frecuencia, respuesta ante escalón y la señal filtrada de la serie de datos ejemplo, comparando los resultados con el ejemplo anterior donde se usó `detrend` . El filtro de Kalman es "causal", esto es, permite ir actualizando el estimado de deriva/tendencia sobre la marcha, en línea, cosa que `detrend` no puede (porque necesita una serie de datos completa hasta el final).

[612: rampkalne]

Eliminación de derivas y rampas (3): filtro de Kalman no
 estacionario

**** 

06:49

Materiales: [[CÓD.:](#) ramp3KalmanNOEstacionario.mlx] [[PDF](#)]

[\[YouTube ►\]](#)


Este vídeo implementa la versión no estacionaria del filtro de Kalman diseñado en el vídeo [611]. Básicamente, excepto en los primeros 30 segundos aproximadamente, después la ganancia coincide con el filtro estacionario, como estaba previsto.

La implementación no estacionaria sólo está justificada en la práctica ante series de datos "cortas"; en la mayoría de aplicaciones el filtro estacionario es más sencillo (lineal invariante en el tiempo), y su salida es casi indistinguible del no estacionario.

El filtro no estacionario debe forzosamente usarse en situaciones donde los patrones de muestreo no sean iguales en todos los instantes (muestreo no convencional). Esa situación se discute en el ejemplo Matlab del vídeo [632] sobre la misma serie de datos, requiriendo modificaciones mínimas al código aquí explicado.

[613: ramprts]

Eliminación de derivas y rampas (4): suavizado
 Rauch-Tung-Striebel no causal

****  06:00

Materiales: [[CÓD.:](#) ramp4RTSsmoother.mlx] [[PDF](#)]

[\[YouTube ►\]](#)

Este vídeo continua el caso de estudio abordado en el vídeo [612]. De hecho, usa los resultados (matrices de varianzas-covarianzas) del filtro de Kalman no estacionario para iterar hacia atrás en el tiempo y reducir la incertumbre del estimado usando las medidas

futuras. Se comparan los resultados del RTS smoother (ver teoría en el vídeo [607]) con los resultados previos sobre los mismos datos (vídeos [610], [611], [612]).


La aplicación del suavizado RTS en escenarios de muestreo no convencional sobre esta misma serie de datos se discute en el vídeo [632].

La salida de un doble integrador ante ruido blanco de entrada en tiempo continuo se conoce como el *Integrated Wiener Process*, y se analiza desde una perspectiva distinta (pero en esencia equivalente) en el vídeo [242], estimando a partir de matrices de varianzas-covarianzas; en ese vídeo también se comprueba que el resultado de la interpolación entre muestras es una *spline* cúbica, esto es, interpolación con funciones que son polinomios de grado tres “piecewise” (por tramos).

17.4.3 Caso 3: ejemplo donde NO funciona bien

[614: kalmal] Comparación observador óptimo (Kalman) versus derivación numérica en sistema de 2o orden: resultados inesperados

Materiales: [[CÓD.:](#) ObserversAreNotPerfect.mlx] [[PDF](#)]

****  14:57

[YouTube ►]

En este vídeo se diseña un controlador por realimentación de la salida (observador + realimentación del estado) para un sistema de 2o orden cuyos estados físicos son la salida y su derivada (posición y velocidad), en tiempo discreto.

El controlador se diseña mediante `dlqr`, y el observador mediante un filtro de Kalman estacionario `dlqe`. Las ganancias de realimentación del estado podrían entenderse como las ganancias proporcional y derivativa de un regulador tipo PD. Como alternativa al observador para estimar la velocidad, se discretiza $s/(s/5+1)/(s/200+1)$ implementando una derivación numérica filtrada paso-bajo, donde las constantes de los filtros se han sintonizado simplemente “a ojo”.

Se simulan ambos métodos de estimar estado (posición y velocidad) ante condiciones iniciales fuera de equilibrio (sin perturbaciones) y se comprueba que producen resultados similares, muy buenos.


Una segunda simulación se hace con condiciones iniciales nulas pero una perturbación constante (punto de funcionamiento mal calculado). Se observa que el observador de Kalman produce un resultado totalmente anti-intuitivo, dando una acción de control en equilibrio con el signo opuesto al que debería. Sin embargo, el filtro “a ojo” sí que funciona como intuitivamente se esperaría.

En resumen, el filtro de Kalman, fuera de las condiciones teóricas de optimalidad que lo justifican, no tiene por qué ser la mejor solución en la práctica ante perturbaciones constantes, no gaussianas, ruido coloreado no tenido en cuenta en el diseño, etc.

17.5 Estimación aproximada en procesos no lineales

[615: normnl] Deformación de distribuciones de probabilidad en sistemas no-lineales: ejemplo Matlab (estático)

Materiales: [[CÓD.:](#) DeformacionDeGausianaNoLineal.mlx] [[PDF](#)]

***  10:45

[YouTube ►]

La distribución normal es ampliamente usada (supuesta) en control y observación lineal (filtro de Kalman) porque las transformaciones lineales de dicha distribución dan lugar a variables aleatorias que tienen también distribución normal.

Por desgracia, cuando el sistema es no-lineal, ese no es el caso. Este vídeo simula tres sistemas (lineal, no-lineal suave, no-lineal con grandes cambios de pendiente) ante una entrada con distribución normal, comprobando el histograma de los resultados y las aproximaciones al mismo obtenidas mediante: (a) el modelo linealizado (distr. normal de salida), o (b) la “unscented transform” (modificación de media y varianza evaluando dos puntos especialmente elegidos con el modelo no lineal). En el caso lineal y no-lineal “suave”, ambas opciones resultan aceptables (en el caso dinámico dan lugar al filtro de Kalman “extendido” y el “unscented” respectivamente); con la no-linealidad abrupta, ninguna de ellas aproxima correctamente al histograma que suponemos “correcto”, determinado con “muchas” tiradas de dado, que en el caso dinámico dará lugar al denominado “filtro de partículas”. Los ejemplos “informales” de este vídeo motivan el estudio de esos tres tipos de generalizaciones del filtro de Kalman lineal, fuera de los objetivos de este vídeo.

17.5.1 Filtro de Kalman extendido

[616: ekfteo] Filtro de Kalman extendido para sistemas no lineales: teoría

****  10:02

Materiales: [FiltrKalmanExt.pdf]

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo plantea algunas consideraciones teóricas que dan lugar a lo que se conoce como el filtro de Kalman extendido. Este filtro es la generalización del filtro de Kalman lineal a sistemas no lineales basada en la linealización (variante en el tiempo) alrededor de la trayectoria del estado estimado (visualizar el vídeo [875] o/y el [877] podría resultar conveniente para afianzar los conceptos sobre este tipo de linealización en los que se basa este filtro).


Básicamente, si el estado “real” x_k está cerca del estado “estimado” \hat{x}_k entonces la linealización de la ecuación de estado permitirá hacer una estimación \bar{x}_{k+1} en media y varianza del valor de x_{k+1} (antes de medir). Análogamente, si x_{k+1} está cerca de \bar{x}_{k+1} la linealización de la ecuación de salida permite aproximar las varianzas y covarianzas de la medida. Con ello, se tiene la información para plantear las ecuaciones estándar del filtro de Kalman. El resultado es un filtro de Kalman no estacionario con matrices de modelo (linealizado) del proceso que cambian según la trayectoria de \hat{x} y de la entrada.

El mayor inconveniente de este filtro es que es una aproximación: con no-linealidades fuertes o condiciones iniciales lejanas al estado real podrían acumularse errores, o incluso ser *inestable*. No obstante, es usado en muchas aplicaciones experimentales debido a su sencillez conceptual, eficiencia computacional, y comportamiento razonablemente bueno con no-linealidades “suaves”.

Un análisis formal del error cometido se aborda en el vídeo [619], como motivación de las modificaciones que darán lugar a lo que se conoce como “unscented” Kalman filter, en otros vídeos de la colección.

Ejemplos Matlab de este filtro aparecen en los vídeos [617] y [618] en una simulación de un péndulo; también puede utilizarse este tipo de filtro en problemas que combinen observación con identificación de parámetros como se aborda en los vídeos [752] y [753].

[617: ekfml1] Ejemplo filtro Kalman extendido (péndulo, I): modelado y linealización Matlab

**  07:23

Materiales: [[CÓD.](#): testEKFconMatlabBuiltIn.mlx] [[PDF](#)]

[YouTube ▶]

Este filtro plantea los pasos iniciales para ejecutar el código de filtro de Kalman extendido que el comando `extendedKalmanFilter` de la Control Systems Toolbox de Matlab implementa.

El modelo que se aborda es un péndulo no-lineal de segundo orden. Se discute el modelado, discretización, y la obtención de jacobianos pasando de numérico a simbólico y viceversa. Los jacobianos evaluados en los puntos de equilibrio darán lugar a la linealización constante utilizada en control lineal; el jacobiano evaluado sobre el estado estimado será usado en el filtro de Kalman extendido, variante en el tiempo.

Los vídeos [618], [627] y [628] prueban los estimadores del estado EKF, UKF y filtro de partículas sobre este modelo de péndulo.

[618: ekfm12]

Ejemplo Filtro Kalman Extendido (II): simulación Matlab y comparación de opciones

Materiales: [[CÓD.:](#) testEKFconMatlabBuiltIn.mlx] [[PDF](#)]

**** 10:52

[[YouTube](#) ►]

En este vídeo se utiliza el modelo no-lineal y los jacobianos (matrices de derivadas parciales) de un péndulo no-lineal de segundo orden cuyo detalle se ha abordado en el vídeo [617].

Aquí se detallan las distintas opciones que el comando `extendedKalmanFilter` de la Control Systems Toolbox necesita:

- El modelo no lineal
- Los jacobianos de la ecuación de estado (`StateTransitionJacobianFcn`) y salida (`MeasurementJacobianFcn`).
- Varianzas de ruido de proceso (`ProcessNoise`) y medida (`MeasurementNoise`)
- Condiciones iniciales del observador en media (`State`) y en varianza (`StateCovariance`).

Con todo ello, iterando los comandos `correct` y `predict` se puede ejecutar el código que implementa el filtro de Kalman extendido.

Para comparar resultados, se comparan las simulaciones temporales de:

1. El filtro de Kalman extendido con modelo no-lineal y jacobiano variante en el tiempo
2. El observador lineal con el modelo linealizado en el origen (jacobiano constante).
3. Un intermedio donde las ecuaciones de medias son evaluadas con el modelo no-lineal pero las ecuaciones de varianza/ganancia del observador utilizan las matrices constantes de la linealización en el origen. Esto es una idea similar a la usada en el vídeo [569]: no hay por qué acumular error si se conoce un modelo no lineal para propagar al siguiente estado, al menos en media.

En este caso de estudio la primera y tercera de las simulaciones obtienen resultados prácticamente idénticos aunque, realmente, la implementación formalmente aconsejada sería la primera.


Este mismo ejemplo es objeto de análisis en el vídeo [627] comparando con un *unscented Kalman filter*, y en el vídeo [628] aplicando un filtro de partículas.

Nota: en este vídeo se ha usado el comando `extendedKalmanFilter` de Matlab. Para lectores interesados en detalle de implementación interno, el vídeo [753] presenta la implementación directa de las linealizaciones alrededor de trayectorias y las ecuaciones del filtro de Kalman no estacionario que, obviamente, también son ejecutadas por los métodos de `extendedKalmanFilter`. De hecho, el referido vídeo utiliza filtro de Kalman extendido para un problema de identificación y observación del estado (podría ser interesante ver en detalle el planteamiento del problema en el vídeo [752]).

17.5.2 Filtro de Kalman-Uhlmann “unscented”

[619: errekf]

Análisis de error del filtro de Kalman extendido (serie de Taylor)


09:50

Materiales: [FiltKalmExtErrores.pdf]


[YouTube ▶]

Este vídeo analiza formalmente las fuentes del error en la propagación de media y varianza de una distribución de probabilidad a través de una función no lineal $y = f(x)$.

La idea básica es que dependerá del desarrollo en serie de Taylor alrededor de la media de la función, y de los momentos de la distribución de probabilidad de x (varianza, sesgo, curtosis). Este análisis inspirará modificaciones (transformación Unscented de Uhlmann-Julier, [620]) para corregir algunos de sus términos.

[620: ukft1]

Transformación unscented (Uhlmann-Julier): idea básica


13:05

Materiales: [UnscentedTransf1.pdf]

[YouTube ▶]


En este vídeo se describe la idea básica detrás de la transformación propuesta por Uhlmann y Julier, que ellos denominaron “unscented”: para evaluar la media y la varianza de $y = f(x)$ se podría reemplazar la distribución de probabilidad de x continua (que requeriría de “muchas” muestras para aproximarse a los resultados correctos) por una distribución discreta \hat{x} con “pocos” puntos (y cuertas probabilidades asociadas a cada uno de los puntos).

Los puntos \hat{x} y sus probabilidades estarían especialmente elegidos para conservar determinados momentos (media, varianza, sesgo, ...) de la distribución original x , de modo que algunos términos de orden superior del error discutidos en el vídeo [619] estuvieran correctamente considerados. En ciertas variaciones de la idea en la literatura, las probabilidades son reemplazadas por “pesos” que pueden incluso tomar valores negativos.

En particular, mantener media y varianza resulta muy sencillo generando $2n + 1$ puntos a partir de la diagonalización de la matriz de varianzas covarianzas $\Sigma = VDV^T$ de la distribución de $x \in \mathbb{R}^n$ original. Estos puntos fueron denominados “sigma-points” por los autores originales de la propuesta. Existe un parámetro ajustable κ relacionado con el peso del punto central (media) en las medias ponderadas de los puntos o sus cuadrados que aparecen en las estimaciones de media y varianza. Otros desarrollos introducen más parámetros ajustables (α, β) , que no son objetivo de este material.

[621: sigptml]

Transformación unscented (generación de sigma-points): ejemplo Matlab


08:23

Materiales: [Cód.: sigmapointgenerar2.mlx] [PDF]


[YouTube ▶]

Este vídeo presenta un breve ejemplo Matlab para generar sigma-points en una distribución unidimensional y en otra bidimensional.

Se observa que los puntos se alejan de la media si aumenta el parámetro κ (peso del punto central); dicho aumento disminuye el peso en las medias ponderadas del resto de puntos. En el caso bidimensional, la diagonalización de la matriz de varianzas-covarianzas produce, además de la media, dos puntos simétricamente distribuidos sobre cada uno de los ejes del “elipsoide de confianza” sobre el que una distribución normal multidimensional agrupa la mayor parte de sus valores (ver vídeo [958]).

[622: ukft2]

Filtro Unscented para sistemas dinámicos no-lineales


11:49

Materiales: [UnscentedTransf2.pdf]


[YouTube ▶]

Este vídeo es continuación del desarrollo teórico de los vídeos [619] y [620]. Se revisan brevemente los conceptos desarrollados en los mismos para pasar a discutir que, considerando las propiedades estadísticas de $y = f(x)$, con $x \sim N(x_0, \Sigma)$, la media y varianza de las imágenes de los puntos de la distribución discreta transformada “unscented” aproximan la media y varianza de las imágenes de la distribución normal original al coincidir varios de sus momentos.

En la segunda parte del vídeo se plantea cómo incorporar la propagación de media y varianza resultante de la transformación “unscented” a un filtro de Kalman, sustituyendo la propagación de medias y varianzas lineales. El resultado se denomina *unscented Kalman filter* (UKF). Existe una variación que se denomina “escalada” que incorpora más parámetros ajustables para intentar mejorar la exactitud, discutida en el vídeo [625].

En bastantes aplicaciones donde las no-linealidades son suficientemente suaves (términos de Taylor de grado 3 despreciables), el UKF sustituye al filtro de Kalman extendido (EKF) debido a su mejor propagación de medias y sencillez computacional. Con no-linealidades abruptas suele ser necesario pasar al filtro de partículas.

[623: ukf1ml] **Propagación de media y varianza en no-linealidades: ejemplo**
Matlab 1D
Materiales: [[CÓD.:](#) unsctest1DnoEscalado.mlx] [[PDF](#)]


***  07:49

[YouTube ▶]

En este vídeo se aborda la estimación de media y varianza de una función $f : \mathbb{R} \mapsto \mathbb{R}$, ilustrando los conceptos teóricos de los vídeos [620] y [622] mediante un sencillo ejemplo Matlab, completando las ideas esbozadas en el vídeo [621] donde sólo se abordaba la generación de sigma-points.

Se comprueban las propiedades de la imagen de los sigma-points, primero en un caso cuadrático y luego se modifica la función incorporando un término cúbico. En el vídeo se comparan las diferencias en la exactitud del estimado entre modelos linealizados, transformada unscented y la estimación mediante la imagen de un millón de “partículas” (teóricamente exacta cuando el número de partículas tiende a infinito).

[624: ukf2ml] **Propagación de media y varianza en no-linealidades: Unscented Transform, ejemplo Matlab 2D**
Materiales: [[CÓD.:](#) unsctest2DnoEscalado.mlx] [[PDF](#)]

***  05:35

[YouTube ▶]

En este vídeo se aborda el mismo problema que en el vídeo [623], esto es la estimación de media y varianza donde intervienen funciones no lineales, pero esta vez usando una función $f : \mathbb{R}^2 \mapsto \mathbb{R}$ como ejemplo en Matlab. Se comprueba primero el caso lineal y luego el cuadrático, comparando las estimaciones de modelos linealizados, unscented y por simulación de múltiples muestras. El vídeo sólo discute ejemplos sin dinámica; un ejemplo de aplicación a un sistema dinámico usando el comando `unscentedKalmanFilter` de Matlab aparece en el vídeo [627].

Transformación unscented escalada

[625: ukfesc] **Transformación unscented escalada**
Materiales: [UnscentedTransfEscalada.pdf]

****  10:43


[YouTube ▶]

En este vídeo se discute la modificación “escalada” de la unscented transform.

En concreto, la motivación es que, según el valor de κ , los sigma-points pueden corresponder a puntos alejados de la media, poco probables. Estos puntos, en no-linealidades


abruptas (no cuadráticas) podrían deformar mucho los resultados numéricos cuando la mayoría de los puntos estarían mucho más cerca de la media. Por tanto, Julier propuso un parámetro de escalado α de modo que las fórmulas de la unscented transform son evaluadas mediante las imágenes de los puntos $\hat{x}^{esc} = x_0 + \alpha \hat{\delta}x$ siendo $\hat{\delta}x$ los incrementos sobre la media x_0 de la unscented transform original.

El parámetro α requiere modificar las fórmulas de cálculo, introduciendo una función auxiliar $\tilde{f}(\delta x) = f(x_0) + \frac{1}{\alpha^2} \cdot (f(x_0 + \alpha \delta x) - f(x_0))$ cuya media y varianza sirven para calcular la media y varianza originales buscadas utilizando los pesos originales de la unscented transform. En la referencia original, sustituyendo \tilde{f} por f se desarrollan expresiones donde la media y varianza son expresadas en términos de la función original f y nuevos pesos dependientes de α que no son desarrollados aquí por brevedad, se remite al lector al vídeo [626] si está interesado en dichos detalles.

[626: ukfesc2] Transformación unscented escalada: reescritura de pesos y eliminación función auxiliar *****  16:16
Materiales: [UnscentedTransfEscaladaReWeight.pdf] [YouTube ▶]

Este vídeo detalla la reescritura de pesos que aparece en las fórmulas finales de la literatura (y documentación de Matlab) donde se ha eliminado la función auxiliar \tilde{f} discutida en el vídeo [625] presentando todos los resultados como medias ponderadas de la función original. La unscented transform escalada reescrita debe alterar el peso que el punto central tiene en la expresión de la varianza respecto al que previamente se ha calculado para estimar la media.

Ejemplo


[627: ekfukfml] Comparación Kalman Extendido/Unscented: ejemplo péndulo no lineal Matlab ***  04:07
Materiales: [Cód.: testEKFvsUKFMatlabBuiltIn.mlx] [PDF] [YouTube ▶]

En este vídeo se discute brevemente la estimación del estado en un modelo de péndulo no-lineal mediante filtro de Kalman unscented, comparando la precisión obtenida con la del filtro extendido ante las mismas medidas. Como los desarrollos son paralelos a los del vídeo [618], aquí sólo se discuten en detalle las tres o cuatro líneas diferentes para implementar el objeto `unscentedKalmanFilter` de la Control Systems Toolbox en vez del `extendedKalmanFilter` del vídeo referido.

17.5.3 Filtro de Partículas

No dispongo de material teórico sobre “particle filter” (simulación Monte-Carlo), que también es muy utilizado en este tipo de problemas cuando las no-linealidades son abruptas o/y las distribuciones de probabilidad usadas son no gaussianas. Se remite al lector a la amplia bibliografía sobre estos temas disponible en otras fuentes.

Una introducción “básica” (en inglés) muy fácil de entender, sin prácticamente fórmulas, aparece en <https://www.youtube.com/watch?v=aUkBa1zMKv4>

[628: pfml] Filtro partículas (control syst. toolbox matlab): ejemplo péndulo no lineal ***  09:55
Materiales: [Cód.: tstEKFvsUKFvsPFMatlabBuiltIn.mlx] [PDF] [YouTube ▶]

Este vídeo discute cómo utilizar el objeto `particleFilter` de la control systems Toolbox de Matlab para estimar el estado de un péndulo no lineal. El modelo de péndulo es el mismo que en los vídeos [617] (que detalla el modelado y linealización), [618] que detalla el filtro de Kalman extendido y el vídeo [627] que detalla el uso del comando `unscentedKalmanFilter`.

Gran parte del código es idéntico al de dichos ficheros, con mínimas modificaciones (se ha incrementado el período de muestreo y se ha integrado con *midpoint* integration, por variar algo). El primer minuto de este vídeo resume las ideas principales de todos estos casos anteriores, pero se remite al lector al detalle de los mismos si así lo necesita.

El vídeo discute los distintos elementos necesarios para construir el filtro de partículas: el modelo (bucle de simulación de ecuación de estado para cada partícula, ecuación de probabilidad de partícula/medida en salida), la inicialización (número de partículas, media, varianza) y la simulación (`predict` , `correct`). Con sólo comentar/descomentar un par de líneas el código está preparado para compararlo con EKF/UKF, que se deja como actividad propuesta al lector.

El filtro es muy usado en robótica (localización con sensores imperfectos):


<https://www.youtube.com/watch?v=mBlsqSI7c1g>

<https://idecona.ai2.upv.es/content/localizaci%C3%B3n-robots-monte-carlo>

Capítulo 18

Estimación y control: escenarios avanzados de muestreo y muestreo no convencional

18.1 Preliminares al control con muestreo no convencional

[629: mnc]	Muestreo no convencional (mutifrecuencia, en red): planteamiento de la problemática Materiales: [MultifrecPlanteamiento.pdf]	***  12:02 [YouTube ►]
------------	---	--

En este vídeo, se plantea en qué consiste un escenario de muestreo no convencional y qué aplicaciones podrían dar lugar a dicho tipo de escenarios. Se argumenta que si los períodos de muestreo implicados son todos suficientemente pequeños para que la degradación de prestaciones sea despreciable respecto a un caso ideal continuo, entonces este escenario no convencional puede ser “transparente” (por ejemplo, un conjunto de PID’s en cascada operando a período pequeño pero diferente cada uno de ellos no necesita ninguna consideración especial en cuanto al muestreo no convencional). En escenarios con redes con cierta latencia, saturación, procesos rápidos, sí que podría ser necesario considerar el muestreo no convencional en el diseño de controladores.

El escenario en el que existen soluciones claras es el que se denomina *multifrecuencia*: se conocen los instantes futuros de actuación y medida (cada sensor y actuador con un período regular) y existe un patrón de muestreo/actuación que se repite indefinidamente. Para este escenario existen soluciones teóricas basadas en representación entrada-salida (función de transferencia) y en representación interna. Las soluciones basadas en representación interna permiten fácilmente control óptimo y observadores multivariados, de una forma que en función de transferencia resulta bastante más compleja, por tanto, como en la mayoría de casos de control multivariable, se recomienda el uso de la representación en variables de estado para abordar el problema multifrecuencia.

Existen otros escenarios de muestreo no convencional donde hay muestreo no periódico, originado por pérdidas o saturación aleatorias de red, variaciones en tiempo de cómputo, o control basado en eventos (donde se mide cuando se alcanza determinado umbral), etc., que no hacen posible predecir con suficiente precisión los instantes futuros de medida o actuación. El problema es más sencillo si los datos circulan con “sello de tiempo” de modo que se puede saber (a posteriori) en qué momento fue tomada una medida o fue modificado un actuador, pero, de todos modos, es un problema complejo que requiere combinar control robusto, estadística, sistemas variantes en el tiempo, etc.

18.2 Control bi-frecuencia (dual rate) por asignación de polos

[630: drt] [Control bifrecuencia en representación interna por asignación de polos](#)

Materiales: [DualRatePlace.pdf]

**** 10:39

[YouTube ▶]

En este vídeo se presenta la metodología de diseño de un regulador con muestreo no convencional (control bi-frecuencia) para el proceso $\dot{x} = Ax + Bu$, $y = Cx$. El período de los sensores se considerará idéntico, igual a T_{sens} y el período de los actuadores será T_{act} . Además, todos los sensores estarán sincronizados (medida en el mismo instante), así como los actuadores.

Bajo estas suposiciones, unas pocas líneas usando `place` permitirán el diseño de observadores y reguladores por realimentación del estado que garantizan estabilidad en bucle cerrado (se cumple principio de separación, aunque no se prueba en este vídeo). Se supone que quien visualice este vídeo tiene claros los contenidos de los materiales sobre asignación de polos en tiempo discreto (monofrecuencia), que se recomienda revisar con antelación a éste.

La implementación deberá hacer código que responda a eventos de (a) “nueva medida”, ejecutando $\hat{x} = \hat{x} + L(y - C\hat{x})$, y (b) de “actualización de control necesaria”, ejecutando $u = -K\hat{x}$. También deberá hacerse simulaciones basándose en la exponencial de una matriz (`expm`) ampliada de la discretización ZOH (se recomienda revisar los materiales del vídeo [371] antes del visionado de este vídeo con la ampliación bi-frecuencia).

[631: drml] [Control bifrecuencia por asignación de polos: ejemplo Matlab](#)

Materiales: [[CÓD.:](#) dualratestatespace.mlx] [[PDF](#)]

**** 16:49

[YouTube ▶]

Este vídeo presenta un ejemplo Matlab del diseño bi-frecuencia cuya teoría se esbozaba en el vídeo [630]: modelado a dos períodos, diseño con `place` a partir de discretización de polos continuos, y simulación con detección de eventos (reloj múltiplo de períodos de medida o actuación), exponencial de matriz para actualizar el estado estimado, corrección del estado y actualización del control.

18.3 Estimación/filtrado con muestras no equiespaciadas

En muchos casos, se dispone de muestras espaciadas de forma irregular y se desea estimar un estado interno o “filtrar” dicha señal.

Modelos entrada-salida. Cuando el problema se aborda desde un punto de vista entrada-salida en estado estacionario (usando autocorrelación, función de transferencia excitada por ruido, power spectral density) entonces los desarrollos se encuadran dentro de la teoría de “procesos estocásticos Gaussianos” y los “filtros de Wiener”. De hecho, los vídeos [krigt(08:44)] (teoría) y [krigtm(10:29)] (ejemplo Matlab) discutidos en el capítulo sobre procesos estocásticos realizan un filtrado (estimación en instantes no medidos) interpolando sobre datos irregulares conocida la función de autocovarianza o, equivalentemente, la densidad espectral de potencia. Por ello, se recomienda al lector que consulte dichos vídeos (y el material introductorio preliminar del capítulo, según sus conocimientos previos) para entender el enfoque del problema de estimación con muestreo no convencional en el marco de los modelos entrada-salida.

Modelos en representación interna. Este tipo de modelos requiere de modificaciones del filtro de Kalman. No dispongo por el momento de los materiales completos para abordar dicho tema en el caso más general (modelo en tiempo continuo muestreado irregularmente, caso *sampled-data*). Si todo se ciñe a tiempo discreto, las modificaciones del filtro de Kalman son bastante sencillas, y la siguiente subsección presenta un caso de estudio en Matlab que ilustra las ideas principales.

18.3.1 Caso de estudio: fusión de sensores a diferente período con filtro de Kalman

[632: rampNC]

Fusión sensorial con muestreo no convencional: eliminación de derivas y rampas Kalman/Rauch-Tung-Striebel

Materiales: [[CÓD.:](#) ramp5KalmanDualSensor.mlx] [[PDF](#)]

13:15

[YouTube ►]

La fusión de datos de navegación inercial (con deriva por la integración de datos de acelerómetros) y de GPS (sin deriva, pero a una frecuencia de muestreo más lenta) es un ejemplo clásico de las posibilidades del filtro de Kalman no estacionario. Este código presenta un ejemplo de dicha fusión sensorial a diferentes períodos de muestreo (de hecho, el período de muestreo será aleatorio) sobre el caso de estudio de eliminación de rampas abordado (para el caso de un único sensor) en los vídeos [610], [611], [612] y [613]. Obviamente, los resultados del filtro de Kalman con doble sensor (y del RTS smoother –teoría en vídeo [607]– si el procesado es fuera de línea) son muy superiores al caso de disponer de un único sensor con deriva.

Nótese que el suavizado RTS para el doble integrador es considerado (de forma cuasi-equivalente) como un problema “Kernel regression” sobre el “proceso de Wiener integrado” en el vídeo [242]; bajo ciertas condiciones, el suavizado producido por un modelo doble integrador es una *spline* cúbica (función que, por tramos, es un polinomio de grado 3 que pasa por las muestras).

Capítulo 19

Control óptimo \mathcal{H}_2 y \mathcal{H}_∞

19.1 Preliminares: normas de sistemas

[633: norms] Normas de señales y sistemas

Materiales: [Norms.pdf]

****  14:58
[YouTube ►]

Este video describe las normas de señales y sistemas dinámicos.

- Revisión de las normas 2 (euclídea), ∞ , norma 1 o norma p de vectores.
- Normas de sistemas lineales estáticos (normas de matrices): normas inducidas.
- Normas de señales deterministas: norma 2 (raíz de integral en el tiempo del cuadrado de la norma 2 en cada instante); norma ∞ (máximo), norma RMS. La *norma 2 acotada* implica que la señal es “no persistente” (tiende a cero cuando $t \rightarrow \infty$), ello servirá para probar estabilidad cuando se hable de norma 2 acotada de salidas de sistemas.
- Señales aleatorias: varianza, power spectral density.
- Norma 2 de un sistema lineal: interpretación estadística (desv. típica ante ruido blanco PSD unidad en caso SISO, o raíz cuadrada de la suma de varianzas en caso de múltiples salidas), determinista (integral resp. impulso), frecuencial (integral resp. frecuencia). Ejemplos Matlab de estas interpretaciones de la norma 2 aparecen en el vídeo [634].
- Norma ∞ de un sistema lineal: entrada “peor caso”, interpretación frecuencial (pico de resonancia).

19.1.1 Interpretación de normas \mathcal{H}_2 y \mathcal{H}_∞ : ejemplos Matlab

[634: norm2ml] Significado de la norma 2 de un sistema LTI: ejemplos Matlab

Materiales: [[Cód.:](#) Norm2Example.mlx] [[PDF](#)]

***  08:54
[YouTube ►]


Este vídeo verifica la coincidencia entre las tres posibles interpretaciones del significado de la norma 2 de un sistema lineal invariante en el tiempo (introducción breve en vídeo [633], desarrollo teórico más completo en vídeo [636]): interpretación temporal (raíz cuadrada de integral de respuesta impulso al cuadrado), interpretación frecuencial (raíz cuadrada de integral de módulo de respuesta en frecuencia al cuadrado) e interpretación estadística (desviación típica – o raíz de la suma de la varianza– ante entrada ruido blanco de densidad espectral de potencia unidad).

Las respuestas se evalúan con `impulse`, `freqresp` y `lsim` en cada caso, respectivamente, y las integrales en los dos primeros son calculadas numéricamente mediante `trapz`.

Se presenta un ejemplo monovariable y otro multivariable (2 entradas y 2 salidas).

[635: ninfml]

Respuesta en frecuencia multivariable y significado de la norma infinito de un sistema LTI: ejemplo Matlab
Materiales: [[CÓD.: NormInfExample.mlx](#)] [[PDF](#)]

***  07:08
 [YouTube ▶]

Este vídeo ilustra las interpretaciones de la norma infinito de un sistema lineal invariante en el tiempo (`norm(G,"inf")`). La descripción introductoria de los conceptos subyacentes se desarrolla en el vídeo [633], y el detalle en los vídeos [637] y [638], pero los últimos vídeos no son necesarios para comprender las interpretaciones “relativamente sencillas” que este vídeo ilustra:


- Interpretación frecuencial (pico de la respuesta en frecuencia, pico del máximo valor singular en caso MIMO)
- Interpretación temporal, como máximo cociente posible entre norma de entrada y salida; se generan las senoides con la amplitud y desfase requeridos para comprobar como el cociente entre las integrales de salida y entrada coincide aproximadamente con lo indicado por el valor de la norma.

Se usan los comandos `norm`, `sigmaplot`, `lsim`, `evalfr` y `frequresp`, entre otros.

19.1.2 Cálculo de la norma \mathcal{H}_2 de sistemas lineales

[636: norm2]

Cálculo de la norma 2 de un sistema lineal
Materiales: [CalculoNorma2.pdf]

*****  10:57
 [YouTube ▶]

Este vídeo demuestra cómo calcular la norma 2 de un sistema, introducida en el vídeo [633] a partir de las matrices A , B , C , $D = 0$ de la representación interna.


El contenido de este vídeo es opcional/especializado: lo importante para aplicaciones de ingeniería de control es la comprensión del significado físico (vídeo [633]), más que las interioridades del comando `norm` de Matlab.

- La integral de la respuesta a impulso se calcula mediante el gramiano de observabilidad (es necesario comprender los conceptos del vídeo [280] antes de visionar éste).
 - La interpretación estadística (desv. típica ante ruido blanco en entrada) posibilita calcular la norma a partir del gramiano de controlabilidad (visionar el vídeo [279] antes que este).
 - *Dualidad:* Dado que cambiando $ss(A, B, C, 0)$ por $ss(A^T, C^T, B^T, 0)$ las matrices/gramianos de controlabilidad pasan a ser de observabilidad y viceversa, ambos cálculos coinciden.
- Ejemplos Matlab de estas interpretaciones de la norma 2 aparecen en el vídeo [634].

19.1.3 Cálculo de la norma \mathcal{H}_∞ de sistemas lineales

[637: ninfa]

Cálculo de la norma infinito de sistemas lineales (I):
 Preliminares
Materiales: [normainfcalc.pdf]

*****  10:37
 [YouTube ▶]

En este vídeo se presentan la motivación y se revisan conceptos preliminares para el resultado principal discutido en el vídeo [638].

En concreto, se revisa el concepto de norma infinito de un sistema lineal (pico de resonancia, máximo en frecuencia del máximo valor singular), y se argumenta que la metodología más sencilla para calcularlo sería hacer un “mallado” con una cierta granularidad en un

determinado intervalo de frecuencia. No obstante, con resonancias “estrechas” podría tener errores significativos, y tampoco está demasiado claro a qué frecuencias buscar. Por ello, se buscan metodologías que eviten el mallado (matriz hamiltoniana, desigualdades matriciales lineales).

El objetivo del vídeo [638] es la metodología de la matriz hamiltoniana. Para ello, se revisan la interconexión en cascada y la inversa de un sistema en representación interna (se pueden consultar los detalles en los vídeos [283] y [284], respectivamente). También se revisa el módulo de números complejos, la norma de vectores de números complejos y los valores singulares de matrices complejas, introduciendo la operación “transpuesta-conjugada”.

[638: ninfb]

Cálculo norma infinito de sistemas lineales (II): Método matriz hamiltoniana

Materiales: [normainfcalc.pdf]

***** 16:26

[YouTube ▶]

Este vídeo es continuación del [637]. Primero, se revisan en un minuto y medio los anteriores preliminares. A continuación, dado un sistema G , se motiva la definición del sistema adjunto $G^\sim = G^T(-s)$, dado que el cuadrado de la respuesta en frecuencia de G viene dado por la respuesta en frecuencia de $\Phi := G^\sim G$.

Dado un cierto $\gamma > 0$, arbitrario y prefijado, la norma infinito de G es menor de γ si y sólo si la de Φ es menor de γ^2 .

Como la respuesta en frecuencia de Φ es real, eso es lo mismo que decir que $\Theta := \gamma^2 I - G^\sim G$ no pase por cero en el eje imaginario. Se argumenta que para que ello ocurra, la respuesta en frecuencia de $\Theta := \gamma^2 I - G^\sim G$ no debe anularse en ningún punto del eje imaginario, lo que significa que el sistema lineal Θ no debe tener ceros imaginarios puros o, lo que es lo mismo, Θ^{-1} no debe tener polos imaginarios puros. Manipulaciones algebraicas permiten obtener la representación interna del sistema lineal Θ^{-1} y su matriz de estado (H), para comprobar si tiene o no autovalores imaginarios puros.

Con este resultado, se puede plantear una optimización por bisección para calcular la norma infinito con la precisión que se desee.

Nota: este vídeo es únicamente de interés para comprender qué realiza el comando `norm` de Matlab, con contenido relativamente difícil y especializado que puede ser saltado sin que suponga inconveniente para la comprensión del resto de conceptos importantes “prácticos” para ingeniería de control en este capítulo.

También, en sistemas *discretos*, la “continuización” mediante la transformación bilineal preserva el *pico* de la respuesta en frecuencia (aunque se produzca a valores diferentes de dicha frecuencia). Por tanto, para calcular la norma infinito de un sistema lineal discreto, puede continuizarse y aplicarse los resultados del vídeo [638] al sistema continuo resultante, sin introducir ningún tipo de error por dicha transformación. Esta continuización también podría aplicarse al diseño de reguladores \mathcal{H}_∞ .

19.2 Planta generalizada

[639: genpl1]

Expresión de problemas de control en forma de planta generalizada

Materiales: [GenPlantOA.pdf]

*** 10:08

[YouTube ▶]

Este video presenta el concepto de “planta generalizada” que engloba muchos problemas de control como casos particulares de un diagrama de bloques (LFT). La idea básica es una generalización del concepto de “rechazo de perturbaciones” del vídeo [571].

- Entradas de la planta generalizada: entradas no manipuladas / entradas manipuladas.
- Salidas de la planta generalizada: objetivos de control (señales a hacer pequeñas) / información al controlador.
- Ejemplo: cancelación de ruido de proceso por realimentación del error, paso a planta generalizada.
- Ejemplo: prealimentación de perturbación medible, paso a planta generalizada.
- Presentación breve de la solución de ese problema (comandos `h2syn` y `hinfsyn`).

Nota: Estos comandos son numéricamente eficientes, para procesos lineales, dando lugar a reguladores de orden usualmente elevado en representación en espacio de estados. La optimización de la norma infinito utilizando reguladores de estructura prefijada se hace bien con el comando `hinfstruct` (que se aborda en el vídeo [845] y en el [834]) o con técnicas genéricas de optimización en, por ejemplo, las cotas en frecuencia del Simulink Response Optimization (el vídeo [588] propone una cota en frecuencia como restricción, pero también puede ser incluida en el índice a optimizar). Por el momento, consultar dichos vídeos puede ser dejado para un vistazo posterior.

Ejemplos de código Matlab construyendo plantas generalizadas aparecen en los vídeos [641], [664], [645] y muchos otros de la colección donde se resuelven casos concretos de control \mathcal{H}_2 y \mathcal{H}_∞ .

El vídeo [642] presenta varias formas de construir plantas generalizadas con la Control Systems Toolbox, aunque si estás comenzando a explorar este tema, quizás sea conveniente que no lo visualices todavía porque llegar hasta el último detalle de implementación puede desviarte de las “ideas conceptuales importantes” que subyacen al concepto de planta generalizada, con el [641] suficiente por el momento.

El concepto de planta generalizada (interconexión LFT) también se puede gastar para separar elementos en tiempo continuo y en tiempo discreto en bucles de control por computador, para simularlos adecuadamente en instantes intermuestreo (ver vídeo [390], por ejemplo).

El detalle de las fórmulas LFT en representación interna se aborda en el vídeo [285], aunque no es necesario para comprender los conceptos aquí.

Nota: Las interconexiones que aparecen en la planta generalizada pueden ser obtenidas/simplificadas a partir de las fórmulas de representación interna de cada uno de sus componentes, como se discute en el vídeo [starp(18:21)]. De todos modos, Matlab lo realiza de forma transparente en sus comandos de interconexión de la control systems toolbox (por ejemplo, `lft`), y los detalles de dichas fórmulas no son necesarios en este momento para la comprensión de los vídeos introductorios de esta sección.

[640: genpl2] [Planta generalizada: caso general multivariable](#)

Materiales: [GenPlant2.pdf]

****  13:35
[YouTube ▶]

Este video complementa las ideas del video [639], presentando un ejemplo completo de problema de control multivariable que combina control con 2 grados de libertad, control indirecto, prealimentación, medida de variables auxiliares (cascada), dinámica de sensores, etc. y lo expresa en forma de planta generalizada. Las manipulaciones asociadas a, por ejemplo, modelos en representación interna, son discutidas en el vídeo [645].

Casos particulares pueden ser el control usual con un grado de libertad –vídeo [649]–, control en cascada con sensor extra –vídeo [656]–, con actuador extra –vídeo [658]–, con dos grados de libertad –vídeo [663]–, o incluso observadores (estimadores de estado, filtros) –vídeo [668]–. *Nota:* estos ejemplos completamente ejecutados en Matlab requieren la comprensión del concepto de planta generalizada ponderada en el vídeo [644], por lo que deberían ser visionados con posterioridad a éste.

Las plantas generalizadas deben ser “ponderadas” para combinar diferentes unidades o prestaciones requeridas en distintas variables, generalizando el concepto de planta “escalada” a ± 1 abordado en la Sección 13.2.

La planta generalizada “ponderada” requerirá, en muchas aplicaciones, la incorporación de filtros paso alto/bajo/banda al modelo físico, por lo que se recomienda revisar el Capítulo 5, y en especial los filtros elementales descritos en el vídeo [filt(09:29)].

También es conveniente, en este momento, tener claro el concepto del análisis en frecuencia de bucles cerrados de control y lo que significan los diagramas de Bode de las distintas funciones de transferencia entre referencias/perturbaciones y acción de control o salidas. Se recomienda, pues, la revisión del vídeo [rfbc(16:46)] o materiales asociados para comprender mejor la generalización de esas ideas que se discute a continuación.

Ejemplo construcción de planta generalizada en Matlab

[641: pgdydu1]

Planta generalizada: ejemplo Matlab rechazo perturbaciones
entrada y salida (Symbolic Toolbox)

Materiales: [[CÓD.:](#) PGbuildDyDu.mlx] [[PDF](#)]

*** 12:48
[YouTube ▶]

Este video plantea cómo crear, usando el Symbolic Toolbox de Matlab, una planta generalizada de un problema de control donde un proceso tiene perturbaciones a la salida (d_y) y a la entrada (d_u), esto es $y = G \cdot (u + d_u) + d_y$. El controlador que se deberá utilizar tiene la estructura $u = -Ky$.

Se comprueba que, una vez construida la planta generalizada, si se cierra el bucle con `lft`, se obtienen las mismas expresiones en bucle cerrado que las que se obtendrían despejando (con el comando `solve`) las variables y , u a partir de las dos ecuaciones arriba indicadas que representan el problema de control bajo estudio.

Aunque este vídeo ha manipulado expresiones simbólicas, en desarrollos completos las plantas generalizadas deberán ser construidas usando código de la Control Systems Toolbox, lo cual se verá en el vídeo [642], de tres formas diferentes. La planta generalizada en este problema se usará, como ejemplo, en un análisis de bucle cerrado con PID's de un proceso con retardo en el vídeo [643].

Otros ejemplos de construcción de plantas generalizadas aparecen, por ejemplo, en el vídeo [651], en un problema parecido al aquí discutido, o el [664] sobre un sistema físico, así como en muchos otros casos de control \mathcal{H}_2 y \mathcal{H}_∞ de esta colección.

[642: pgdydu2]

Planta generalizada: ejemplo Matlab rechazo perturbaciones
entrada y salida (Control Toolbox)

Materiales: [[CÓD.:](#) PGbuildDyDu.mlx] [[PDF](#)]

**** 18:37
[YouTube ▶]

Este vídeo es continuación del [641], donde se planteaba el diagrama de bloques de un rechazo de perturbaciones simultáneo a la entrada y a la salida, y se cerraban los bucles con `lft` operando sobre objetos de la Symbolic Toolbox.

En control multivariable, debe operarse con objetos `ss`, `tf`, `zpk`, ... de la Control System Toolbox y de la Robust Control Toolbox; por tanto, el objetivo de este vídeo es plantear tres posibilidades de construir plantas generalizadas mediante la Control Toolbox: (a) En Matriz de Transferencia, (b) en Representación interna, (c) en Diagrama de Bloques mediante el comando `connect`.

Cada una de ellas tiene sus ventajas e inconvenientes. Se hace énfasis en evitar la multi-incidencia (escribir varias veces el modelo G o las matrices A , B , C , D del mismo) para que las toolboxes de Control no se encuentren con “duplicados” de sistemas o parámetros inciertos de los mismos.

La planta generalizada en este ejemplo se usará en un análisis de bucle cerrado con PIDs de un proceso con retardo en el vídeo [643].

Otro ejemplo (parecido a éste) de varias formas de construir una planta generalizada aparece en el vídeo [665], sobre un modelo de un horno.

El detalle interno de cómo interconectar sistemas en LFT se discute en el vídeo [285].

[643: pgdydu3] **Planta generalizada: análisis rechazo perturbaciones entrada y salida usando reguladores PID**

Materiales: [[CÓD.:](#) PGbuildDyDuANALISISBC.mlx] [[PDF](#)]

*** 14:31

[YouTube ▶]

Este vídeo utiliza la planta generalizada de un problema de rechazo de perturbaciones en entrada y salida para evaluar las prestaciones de un par de reguladores PID sobre un proceso con retardo. La construcción de la planta generalizada se resume en los dos primeros minutos, ver los detalles discutidos ampliamente en los vídeos [641] y [642].

Una vez construida la planta generalizada, se cierra el bucle con `lft`, que resulta más cómodo que cuatro líneas con diferentes opciones del comando `feedback`, una alternativa SISO que sería la que se explicaría en un curso de grado.

Se presentan y analizan los polos en bucle cerrado (usando aproximación de Padé), la respuesta temporal y la respuesta en frecuencia.

19.2.1 Planta generalizada ponderada para control óptimo

[644: pgpo] **Planta Generalizada Ponderada**

Materiales: [GenPlantWeighted.pdf]

**** 19:09

[YouTube ▶]

En este vídeo se discuten las ponderaciones necesarias para plantear de forma correcta una aplicación de control en el marco de “planta generalizada” de modo que las soluciones \mathcal{H}_2 o \mathcal{H}_∞ tengan sentido físico. Es una generalización multivariable de los conceptos sencillos iniciales considerados, por ejemplo, en los vídeos [583] y [586].

- Ponderaciones de variables controladas: estáticas (escalado, márgenes de error), dinámicas (tiempo de establecimiento, ancho de banda).
- Ponderaciones de perturbaciones: ruido blanco/coloreado... por ejemplo, el filtro generador de ruido coloreado considerado en el caso de estudio del vídeo [220].
- Ponderación de referencias: amplitud, ancho de banda.
- Ponderaciones de variables manipuladas (actuadores): amplitud hasta saturación, ancho de banda.
- Construcción de la planta generalizada ponderada.
- Conclusiones: tipos de filtros usuales en las entradas y salidas generalizadas.

[645: pgposs] **Planta generalizada: operaciones en representación interna**

Materiales: [GenPlant3ss.pdf]

*** 09:31

[YouTube ▶]


Este vídeo plantea cómo manipular las ecuaciones de una planta física en representación interna cuando forma parte de una planta generalizada en un problema de control multivariable según discutido en el vídeo [640].

Primero repasa en los dos primeros minutos las ideas principales de dicho vídeo y después, dando nombres a las distintas variables empleadas, pasa a formar las ecuaciones de estado

y de salida de la planta generalizada (no ponderada). En la parte final del vídeo se discute la idea que, una vez formada la planta generalizada ponderada, la dinámica añadida por los filtros de entrada es no controlable, mientras que la añadida por los filtros de salida es no observable (desde las entradas y salidas al controlador, respectivamente).


Un ejemplo de cálculo de planta generalizada (no ponderada) usando esta metodología aparece en el vídeo [642] (quizás debas visionar el [641] antes); otro ejemplo en un sistema físico (térmico) aparece en el vídeo [664], y otro en el [665].

El vídeo [285] describe la interconexión LFT de dos sistemas en representación interna.

[646: h2hinfb]	Control H2/Hinfinito, condiciones para problema bien planteado	****  10:01
Materiales: [MatVideoh2hinfb.pdf]		[PLAY ►]

En este vídeo se resume lo que busca el control óptimo \mathcal{H}_2 y \mathcal{H}_∞ , y se revisan las condiciones que se necesitan para que el problema esté bien planteado: – significado físico de las condiciones “formales” de la planta generalizada sobre: estabilizabilidad, detectabilidad, polos y ceros en eje imaginario, matriz D .

Errata: Los subíndices D_{12} , D_{21} están “bailados” en la transparencia 58 ítem 3, minuto 06:05.

[647: h2lqrml]	Equivalencia h2syn con (dlqr+dlqe): ejemplo Matlab	*****  27:21
Materiales: [Cód.: h2vslqr2m17b.mlx] [PDF]		[YouTube ►]


En este vídeo se explica el código Matlab para transformar:

- un **problema LQR** (minimizar la integral de $J = z^T Q_z z + u^T R u$, con $z = C_1 x$) + observador óptimo (mínima varianza de error de predicción dadas varianzas de ruidos de proceso y de medida) en un problema de **planta generalizada ponderada**.
- Un problema de **planta generalizada** (con ponderación unidad dado que los escalados/ponderaciones ya se suponen implícitos en la planta generalizada, esto es, minimizar varianza de salidas y_1 cuando las varianzas de entradas exógenas son 1) en un problema **LQR+LQE**.

Esto es, las dos formulaciones son equivalentes. El problema de planta generalizada es, por tanto, una reformulación con “pesos en frecuencia” de las ideas de ampliar el estado con “generadores de perturbaciones” discutidas en el vídeo [576]. Los generadores de perturbaciones no controlables (pero sí observables) corresponderían a la representación interna de los pesos de las entradas a la planta generalizada; los pesos en frecuencia de ciertas salidas corresponderían a ampliar la planta con modos controlables pero no observables (que, por brevedad, no se discutieron en dicho vídeo).

Nota: Un caso más sencillo, donde sólo se discute un ejemplo Matlab de la equivalencia entre el filtro de Kalman estacionario en tiempo discreto y el “observador \mathcal{H}_2 ” aparece en el vídeo [668]–.

19.3 Ejemplos de código

[648: gpex1]	Control planta generalizada H2/Hinfinito (I): ejemplo Matlab/Octave 1 grado libertad	****  14:55
Materiales: [Cód.: ejplg1gl.zip] [PDF]		[YouTube ►]

Este video discute la cancelación de una perturbación d sobre una salida $y = Gu + d$, con una medida m corrompida por ruido: $m = y + r_m$.

Se construye la planta generalizada, y se añaden pesos para construir la planta generalizada ponderada.

Se programan en GNU Octave (código compatible con Matlab) las optimizaciones \mathcal{H}_2 y \mathcal{H}_∞ y se presentan ambas soluciones. En Octave, se necesita el paquete `control`, y en Matlab el *Robust Control* toolbox. Este código, no obstante, ejecutó correctamente en Octave 4.1, con el que se grabó el vídeo.

Se ilustran las respuestas en frecuencia de salida y acción de control, así como la respuesta ante escalón.

Nota: si los pesos de las entradas generalizadas (W_{in} , W_{ruidom}) son pequeños a alta frecuencia, entonces la ganancia del regulador $hinf_{syn}$ puede ser alta; por ejemplo, el ruido de medida podría llegar a saturar actuadores, eso produce gran VARIANZA o poca tolerancia a errores de modelado en el regulador H-infinito. En muchos casos, se aumenta el tamaño de W_{in} o W_{ruidom} a alta frecuencia aunque físicamente sepamos que no va a ser tan grande en las aplicaciones para limitar la ganancia del actuador. También se puede (o se debe en muchos casos) limitar u a altas frecuencias haciendo W_u igual a un filtro paso ALTO. Todos esos detalles no son objeto de este material introductorio, pero deben ser tenidos en cuenta para diseñar correctamente los controladores H-infinito; véase, por ejemplo, el vídeo [653] para un análisis de todo ello.

De hecho, al hilo del párrafo de arriba, existen otras opciones para problemas parecidos (bueno, casi equivalentes al menos en el caso SISO) donde en vez de r_m se considera contenido de alta frecuencia en d que no se desea cancelar, en concreto el enfoque *mixed sensitivity*; ver, por ejemplo, el vídeo [812] (teoría) y el ejemplo Matlab en [816].

[649: ef1] [Control planta generalizada H2/Hinfinito \(II\): codificación Mixed Sensitivity \(Planteamiento\)](#)

Materiales: [[CÓD.:](#) ef1y2.zip] [[PDF](#)]

**** 16:26

[YouTube ▶]

En este video se plantea un problema prácticamente igual al del video [648], pero se simplifica poniendo pesos sólo en salida y acción de control, justificando los pasos. En este video se discute el planteamiento del problema y el código Matlab/Octave para resolverlo. Un proceso y filtros concretos serán evaluados en el video [650]. De hecho, el problema planteado puede ser un caso del problema de sensibilidad mixta (*mixed sensitivity* en literatura en lengua inglesa) que se discute en el video [811] una vez se han introducido consideraciones de robustez (es necesario complementar la teoría para justificar que aumentar la ponderación de la acción de control aumenta la tolerancia a errores de modelado, que se discute en el video [799]).

[650: ef2] [Control planta generalizada H2/Hinfinito \(III\): Mixed Sensitivity \(Efecto de filtros\)](#)

Materiales: [[CÓD.:](#) ef1y2.zip] [[PDF](#)]

**** 20:58

[YouTube ▶]

Este vídeo es continuación del vídeo [649], donde se planteaba el problema de rechazo de perturbaciones para una planta G genérica. En este video, se usa $G = 3/(s^2 + 0.25s + 1)$, y se diseñan los filtros de primer orden adecuados para plantear una planta generalizada ponderada que refleje unos objetivos realistas de control. Se presentan las soluciones \mathcal{H}_2 y \mathcal{H}_∞ .

Se comparan las respuestas (en frecuencia y temporal ante escalón) de error y acción de control. También se compara con dos ponderaciones de error distintas, para comprender el significado de dichos filtros de ponderación.

19.3.1 Caso de estudio: compensación perturbaciones entrada

[651: hiid] **Control H-infinito para rechazo de perturbaciones a la entrada:**
ejemplo Matlab (pesos constantes)

Materiales: [[CÓD.:](#) hinfinputdisturbanceSimple.mlx] [[PDF](#)]

**** 12:52

[YouTube ▶]

Este vídeo presenta el control \mathcal{H}_∞ para un problema de rechazo de perturbaciones a la entrada, $y = G \cdot (u + \delta_u)$ en un proceso lineal de orden 3, de una entrada y una salida. La medida de y (información al controlador) está corrompida por un ruido de medida ρ_m , esto es, $y_{med} = y + \rho_m$.

Se plantea la planta generalizada que codifica el problema (se aconseja visualizar el vídeo [641], donde se presenta la planta no ponderada para un problema muy similar). Luego, se proponen pesos constantes denotando la amplitud máxima de las perturbaciones exógenas δ_u y de ρ_m , y como objetivos de control se propone minimizar la norma infinito de y incorporando asimismo un límite (constante) de saturación de u mediante los adecuados pesos de las salidas generalizadas (inversas de los límites buscados).

La parte final del vídeo analiza las prestaciones en bucle cerrado resultantes evaluando la respuesta temporal (step) y en frecuencia (sigma, bodemag) de `lft(PlantaGeneralizada,K)*Win`. Se comprueba que existe una excesiva sensibilidad a ruido de medida que, por tanto, hará poco realista este diseño para aplicaciones; ello requeriría modificaciones en los pesos (o/y optar por un diseño \mathcal{H}_2 si se desea mínima varianza), no cubiertas en el presente vídeo.

La sensibilidad a ruido de medida (excesiva) y las posibles soluciones con pesos que varían en la frecuencia son analizadas en el vídeo [653], continuación de éste.

Un ejemplo de implementación en tiempo discreto (siempre debería acabarse implementado por computador) de este mismo problema se discute en el vídeo [675]. La implementación práctica debe llevar antiwindup para evitar problemas por saturación, como se discute en el vídeo [676].

[652: h2id] **Control óptimo H2 rechazo perturbación a la entrada: diseño y simulación ante ruido**

Materiales: [[CÓD.:](#) h2inputdisturbance.mlx] [[PDF](#)]

**** 12:12

[YouTube ▶]

Este vídeo plantea la solución \mathcal{H}_2 de un problema de rechazo de perturbaciones a la entrada cuyas soluciones \mathcal{H}_∞ se detallan en los vídeos [651] (pesos constantes) y [653] (pesos en frecuencia).

En este caso \mathcal{H}_2 se hace énfasis en la simulación ante ruido en tiempo continuo con la densidad espectral de potencia dada por los pesos de entrada (mediante discretización aproximada y escalado inversamente proporcional a la raíz cuadrada del período de muestreo), para acercarse a lo que realmente \mathcal{H}_2 trate de optimizar (variación total ponderada en la frecuencia) y no las respuestas “escalón” o “senoidal” de un análisis determinista.

[653: hiid2] **Control H-infinito para rechazo de perturbaciones a la entrada:**
pesos en frecuencia y simulación ante ruido

Materiales: [[CÓD.:](#) hinfinputdisturbance.mlx] [[PDF](#)]

***** 15:13

[YouTube ▶]

Este vídeo plantea el mismo problema que el vídeo [651], con un mínimo cambio en la cota en frecuencia de la perturbación. En este vídeo, además de los pesos constantes (estudiados en el referido vídeo [651], de contenido más sencillo que éste para un estudiante) se estudian otras opciones de pesos en frecuencia. Como es estándar en \mathcal{H}_∞ , se simulan ante escalón y se calculan los diagramas de respuesta en frecuencia en bucle cerrado.

Pero, como detalle importante, también se simulan ante ruido blanco los resultados del control \mathcal{H}_∞ , pese a que formalmente si ese fuera el problema debería haberse usado \mathcal{H}_2 . Se observa cómo los pesos en frecuencia pueden aproximar el resultado al del \mathcal{H}_2 , disminuyendo la varianza debido al ruido de medida.

La parte final del vídeo plantea la posibilidad de aumentar “artificialmente” una señal exógena de entrada (en este caso ruido de medida) incluso a amplitudes mayores que las esperadas “físicamente”, para disminuir la ganancia del regulador (y, por tanto, la varianza ante dicha señal cuando sea menor a la que el optimizador supuso).

En efecto, una conclusión importante es que \mathcal{H}_∞ no es ni mejor ni más robusto que \mathcal{H}_2 “per se” si no se incorporan adecuadamente conceptos de control robusto que no son objetivo de este material. De hecho, la diferencia en sensibilidad (varianza) de la acción de control a ruido puede ser muy grande respecto al (teóricamente óptimo) \mathcal{H}_2 .

En general *si las entradas generalizadas tienen amplitud pequeña y no se toman las precauciones adecuadas, la ganancia del regulador óptimo \mathcal{H}_∞ será excesivamente alta con lo que la sensibilidad a ruido de medida y errores de modelado hará que el diseño no funcione en la práctica.*

Un ejemplo de implementación en tiempo discreto (siempre debería acabarse implementado por computador) de este mismo problema se discute en el vídeo [675]. La implementación práctica debe llevar antiwindup para evitar problemas por saturación, como se discute en el vídeo [676].

19.3.2 Selección de pesos con plantas previamente escaladas

Nota: Los vídeos de esta sección requieren la comprensión previa, entre otros, de los conceptos discutidos en los materiales de los vídeos [sa3ref(10:26)], [dsvd1(12:18)] y [dsvdml(15:45)].

[654: hinfeml]

Selección de pesos para control H-infinito: ejemplo plantas escaladas a +1/-1 (SISO)

****  19:52

Materiales: [[CÓD.:](#) hinfescaladoSISO.mlx] [[PDF](#)]

[[YouTube](#) ►]

En este vídeo se plantea el control \mathcal{H}_∞ para seguimiento de referencias, de la planta mono-variable $G_{esc}(s) = 4/(s+1)/(0.11s+1)$, suponiendo que ya ha sido escalada previamente, de modo que se desean incrementos de salidas en el rango ± 1 , con entradas en el rango ± 1 , saturando fuera de dicho rango; el escalado se discute en detalle en el vídeo [433].

Ello facilita el diseño de los pesos, dado que los pesos de referencias y actuadores pueden fijarse a la constante 1, y el peso del error de bucle puede tener un *ancho de banda* (frecuencia cruce de 0 dB) cercano al ancho de banda de G_{esc} . Aunque luego se hacen ligeras modificaciones, el escalado previo hace que la sintonía inicial de pesos a 1 resulte en un controlador muy cercano a la solución finalmente diseñada.

Numéricamente, se comprueba que al reducir la “tolerancia” del valor óptimo de la norma ∞ en bucle cerrado, algunos de los polos de bucle cerrado tienden a infinito. Por cuestiones de robustez numérica del resultado, Matlab propone por defecto una tolerancia del 1%.

Se comprueban respuestas temporal ante escalón, y en frecuencia, del bucle cerrado resultante, tanto de salida como de acción de control.

Finalmente, se reduce el orden del regulador mediante reducción equilibrada *ponderada en frecuencia* (enfaticando ajuste en el intervalo de cero a 50 rad/s) para evitar que la adición de pesos en el problema de planta generalizada suba el orden y complique demasiado el regulador finalmente propuesto.

Nota: Aunque los pesos han sido aquí conformados con una fórmula de primer orden basada en error de posición y ancho de banda, el comando `makeweight` realiza la misma tarea, con lo que se aconseja ver la documentación Matlab del mismo.

[655: hinf2ml] Selección de pesos en control H-infinito: plantas escaladas multivariantes, enfoque SVD desacoplado
Materiales: [[Cód.:](#) hinfescaladoSVD.mlx] [[PDF](#)]

***** 17:58
 [YouTube ▶]

Este vídeo generaliza la idea del vídeo anterior [654] al caso multivariable. Se presenta una planta escalada 2×2 con ganancias máxima y mínima por encima de uno (ver vídeo [433] para detalles), pero con unos anchos de banda en las maniobras de máxima/mínima ganancia marcadamente diferentes (frecuencias de cruce por 0 dB alejadas).

Al intentar controlar las dos salidas para seguir referencias arbitrarias con una misma velocidad, sólo se puede conseguir un ancho de banda muy lento. Sin embargo, una casi insignificante reducción en el ancho de banda objetivo de la maniobra “difícil” (de poca ganancia) permite acelerar las prestaciones de las maniobras “fáciles” (alta ganancia).

Con ello, se establece una conexión importante entre la selección de pesos en control \mathcal{H}_∞ (para seguimiento de referencias), los escalados discutidos en el vídeo [433] y el desacoplamiento SVD de los vídeos [489], [497] o [498].

Obviamente, en un proceso genérico, los cambios de variable basados en el SVD de la matriz de ganancia estática (frecuencia *cero*) *no* son coincidentes con las direcciones de máxima y mínima ganancia a las *frecuencias límite* del ancho de banda, y la presencia de perturbaciones (ruido de proceso) también cambia la direccionalidad de las maniobras: la metodología no es totalmente “exacta”, pero serviría para acelerar las prestaciones de un diseño basado en la sintonía “manual” de PID’s sobre desacoplamiento SVD, a costa, claro, de complicar bastante (orden elevado, metodología \mathcal{H}_∞ mucho más especializada) el regulador resultante.

Un enfoque basado en ideas similares a las aquí presentadas, en un contexto de control óptimo \mathcal{H}_2 se desarrolla en el vídeo [666]. También, cuando los límites en la acción de control vienen impuestos por la tolerancia a errores de modelado también pueden aplicarse las ideas de este material, como se discute en el vídeo [820].

19.3.3 Ejemplo de aplicación: control en cascada (sensor extra)

En esta subsección, se discute la comparación de dos estructuras de control en una planta $y = G_2 y_1 + p$, $y_1 = G_1 u + \delta$:

- Control con sólo un sensor de y
- Control con sensores tanto de y como de y_1 .

La segunda de las opciones, si G_1 tuviera una dinámica más rápida que G_2 correspondería al caso típico de control en cascada con sensor extra. Con separación de escalas de tiempo, podrían sintonizarse bucles “maestro” y “esclavo” de forma independiente. Esta estrategia de control en cascada es muy común en la práctica, discutida en el vídeo [ectcas(19:03)], incluso en entornos sin modelos, con sintonía “manual” de PID’s.

El objetivo de los dos vídeos de esta sección es ilustrar que, con las suposiciones adecuadas, la ganancia de prestaciones obtenida al incorporar ese sensor extra puede ser justificada en el marco del control \mathcal{H}_2 y \mathcal{H}_∞ sobre plantas generalizadas. Obviamente, los contenidos de los vídeos [cash2(11:29)] y [cash1(10:59)] no pretenden justificar que esta sea la opción más recomendable en la práctica en un caso general: con toda probabilidad, sería el sintonizar 2 PID’s manualmente, pero en casos de sistemas complejos/multivariantes o sin una separación clara de escalas de tiempo, la opción de control óptimo en esta sección podría ser considerada.

[656: cash2] Control en cascada (planta gen): ejemplo Matlab, diseño \mathcal{H}_2
Materiales: [[Cód.:](#) CascadaH2.mlx] [[PDF](#)]

***** 11:29
 [YouTube ▶]

Este vídeo considera un ejemplo de control óptimo \mathcal{H}_2 en un proceso donde si tuviéramos PIDs haríamos control en cascada, con dos sensores asociados a diferentes dinámicas $G_1 = 10/(s + 10)$ (subsistema rápido), $G_2 = 6/(s + 1)/(0.25s + 1)$ (subsistema lento). Ante determinadas suposiciones sobre el contenido en frecuencia de las perturbaciones, renunciar al sensor de y_1 implica obtener una varianza de salidas un 25% mayor (a bajas frecuencias, enfatizadas con los adecuados filtros a las salidas de la planta generalizada ponderada) ante los mismos ruidos coloreados como entradas de perturbación, comparada con el caso en el que se dispone de dos sensores.

El vídeo [657] aborda el problema con optimización \mathcal{H}_∞ .

[657: cashi] Control en cascada (planta gen): ejemplo H-infinito Matlab **** 10:59
 Materiales: [Cód.: CascadaHinf.mlx] [PDF] [YouTube ►]

En este vídeo se considera la versión \mathcal{H}_∞ del mismo problema que el vídeo [656] (lo que, en control descentralizado, se resolvería con control en cascada). En este caso se busca el regulador más rápido (mayor ancho de banda de un filtro paso-bajo sobre la salida controlada) que no sature la acción de control suponiendo que las perturbaciones son senoidales o mezclas de ellas de modo que su amplitud a cada frecuencia está, como máximo, por encima de unos umbrales conocidos (que conforman los pesos de las entradas de la planta generalizada). Como resultado de la optimización, el sensor adicional permite duplicar la rapidez garantizada por el diseño \mathcal{H}_∞ comparado con el control \mathcal{H}_∞ con un sensor, ante unas cotas en frecuencia de perturbaciones idénticas.

NOTA: El diseño es MUY sensible al ruido de medida de alta frecuencia en el sensor auxiliar δ , se ha hecho (exageradamente) demasiado rápido y la acción de control amplificaría mucho dicho ruido o/y originaría una gran varianza de bucle cerrado. Debería rehacerse el diseño subiendo la amplitud a alta frecuencia de δ o/y limitando la actividad a alta frecuencia del actuador.

19.3.4 Ejemplo de aplicación: control en cascada con actuador extra


[658: dacthinf1] Control H-infinito con 2 actuadores de diferente ancho de banda **** 08:36
 (ejemplo Matlab): planteamiento del problema [1]
 Materiales: [Cód.: cascadaSobraActHinf.mlx] [PDF] [YouTube ►]

Este vídeo considera un diseño mediante la metodología de planta generalizada **hinfsyn** del proceso con dos actuadores cuya controlabilidad entrada-salida fue discutida en el vídeo [445], y sobre el que se diseñaron unos PIDs con la estructura de control en cascada con actuador extra, con objetivos muy similares a los que aquí se abordan, en el vídeo [509], para poder hacer análisis comparativo entre ambas opciones.

En este vídeo se discute el planteamiento del problema (la simulación de resultados se detallará en el vídeo siguiente [659]). En particular, se revisa en un minuto (detalles en vídeo [445]) la controlabilidad entrada-salida SVD (**sigma**) y se plantea la planta generalizada no ponderada. Luego, se discuten los distintos filtros presentes en la planta generalizada ponderada:

- la ponderación del error, con un filtro $W_{err} = \frac{80}{\frac{80}{bw}s + 1}$ cuya inversa genera una plantilla de error que indica que el control debe ser efectivo (sensibilidad por debajo de 0 dB) a frecuencias menores a bw , que se interpreta pues como un “ancho de banda” de control;
- La ponderación W_{in} de la referencia de entrada, de valor 1 a alta frecuencia (que será controlada lo más rápido que se pueda maximizando bw en W_{err}), pero de posible mayor amplitud a frecuencias reducidas dado que el análisis de controlabilidad entrada-salida SVD así lo permitía;


- La ponderación de la acción de control. Los actuadores saturan a 1, con lo que esa es la máxima amplitud de su respuesta en frecuencia. Se plantean pesos diferentes para cuatro casos:
 1. Sólo actuador 1, $W_u = \text{diag}(1, 10^4)$;
 2. Sólo actuador 2, $W_u = \text{diag}(10^4, 1)$;
 3. Dos actuadores igualmente ponderados, $W_u = \text{diag}(1, 1)$;
 4. Dos actuadores, pero requiriendo que el actuador 1 retorne a su punto de funcionamiento en maniobras a baja frecuencia para no perder “capacidad de respuesta” a cambios rápidos (básicamente, equivalente a la acción integral en el bucle lento “esclavo” del control en cascada con actuador extra del vídeo [509]).

[659: dacthinf2] **Control H-infinito con 2 actuadores de diferente ancho de banda** ****  11:19
 (ejemplo Matlab): simulación de resultados [2]
Materiales: [[CÓD.:](#) cascadaSobraActHinf.mlx] [[PDF](#)] [[YouTube](#) ►]

En este vídeo se ejecuta el código y se simulan los casos 3 y 4 de control óptimo \mathcal{H}_∞ cuyo planteamiento se discutió en el anterior vídeo [658]. Se presenta respuesta en frecuencia de bucle cerrado, así como simulación de respuesta temporal ante escalón y rampa. Se observa que ambos casos producen resultados similares en respuesta temporal a los de los PIDs del vídeo [509], en particular el caso 4.

Como conclusión de los ejemplos de estas secciones 19.3.3 y 19.3.4, una selección adecuada de los pesos permite replicar las ventajas de sensores y actuadores extra en estructuras de control multi-variable replanteando el problema como un control óptimo en planta generalizada.

19.3.5 Prealimentación (feedforward)

[660: fhinf] **Prealimentación (feedforward) usando optimización H-infinito** ****  13:51
Materiales: [[CÓD.:](#) HinfFeedForwardSiso.mlx] [[PDF](#)] [[YouTube](#) ►]

Este vídeo plantea un problema de prealimentación (calcular u en $y = Gu + G_{pert}d$ siendo d medible), mediante \mathcal{H}_∞ . Para una revisión de los conceptos básicos del feedforward en modelos lineales en un marco clásico (sin optimización), véanse los vídeos [475] (teoría), o ejemplos en [525], [480] o [481].

En este caso G es de fase no mínima (esto no es especialmente relevante aquí, al optimizador le “da igual”; obviamente no se pueden cancelar elementos de fase no mínima porque el regulador ideal $G^{-1}G_{pert}$ sería inestable, pero eso ya lo tiene en cuenta la teoría que justifica `hinfsyn`).

La planta generalizada es $[G_{pert} \ G; 0 \ 1; 1 \ 0]$, y se proponen pesos adecuados sobre y y sobre u para conseguir la mejor cancelación del efecto de $G_{pert}d$ sobre y (enfaticando rechazo de perturbaciones a bajas frecuencias), pero compatible con limitaciones sobre la ganancia de $F(s)$ en $u = F(s)d$, así como la amplitud de u . También se ponen pesos sobre el contenido en frecuencia de d , suponiendo que a altas frecuencias su contenido frecuencial es un 5% como mucho que el de baja frecuencia.

Se analizan en tiempo y frecuencia las prestaciones obtenidas.

El mismo problema, pero incorporando retardos en G y G_{pert} se aborda en el vídeo [661], de hecho, allí se usan otros parámetros en el peso sobre la salida y para cambiar su forma; se aconseja la visualización de dicho vídeo tras este.

[661: fhinfd]

Prealimentación (feedforward) usando H-infinito (caso con retardo)

Materiales: [[Cód.:](#) HinfFeedForwardRetardo.mlx] [[PDF](#)]

10:59

[\[YouTube ►\]](#)

Este vídeo plantea un problema de control feedforward, prealimentación, (calcular u en $y = Gu + G_{pert}d$ siendo d medible), mediante \mathcal{H}_∞ , en presencia de retardo. Se abordan tres casos:

1. Sin retardo (entonces el problema es el mismo que en el vídeo [660] cuya visualización previa es muy aconsejable).
2. Con retardo en la planta G , con lo cual el problema es “difícil” dado que durante el retardo de G no se puede cancelar nada del efecto de G_{pert} . Un enfoque “manual” a un problema básicamente igual, sin usar \mathcal{H}_∞ , aparece en el vídeo [481].
3. Con retardo en la perturbación, con lo cual el problema es más “fácil” dado que medimos la perturbación anticipadamente y podemos “prepararnos” adecuadamente para cuando llegue el “golpe”.

La planta generalizada para hinfsyn es `pade([Gpert G;0 1;1 0])`, porque los sistemas con retardo puro no pueden ser tratados por dicho código. Para simulación, se cerrará con la planta sin la aproximación de Padé.

Los pesos propuestos son los mismos que en el vídeo [660] antes referido, aunque con modificaciones en sus parámetros de errores residuales, anchos de banda, etc. porque las prestaciones conseguibles dependen del retardo.

Se analizan en tiempo y frecuencia las prestaciones obtenidas en cada uno de los tres casos.

19.3.6 Ejemplo de aplicación: control con 2 grados de libertad

[662: hi2gls]

Control con 2 grados de libertad (H-infinito) para seguimiento de referencias: ejemplo Matlab

Materiales: [[Cód.:](#) hinf2glSimple.mlx] [[PDF](#)]

11:59


[\[YouTube ►\]](#)

Este vídeo presenta el diseño \mathcal{H}_∞ de un control de 2 grados de libertad para seguimiento de referencias en un proceso **inestable**, comparando la estrategia 2GL $u = K_e e + K_r r$ con la estrategia 1GL estándar $u = K_{1GL} e$. Como resultado, el ancho de banda de reducción de error en 2GL es aproximadamente cuatro veces superior al 1GL... aunque no es oro todo lo que reluce:

En efecto, la parte final del vídeo ilustra que al haber intencionalmente optimizado la respuesta ante referencia, la respuesta ante perturbaciones es mucho peor que el diseño 1GL, dado que la planta generalizada 2GL no ha tenido en cuenta dichas prestaciones. Un análisis correcto de la presencia simultánea de cambios de referencia y perturbaciones requiere modificaciones a la planta generalizada, discutidas en el vídeo [663], continuación de éste.

Por otro lado, si el proceso fuera estable y de fase mínima, las prestaciones de seguimiento de referencia serían IDÉNTICAS para 1GL, 2GL (o incluso para bucle abierto, $\text{Planta-GenBA} = \text{PlantaGen}([1 \ 2 \ 4];:)$). Por eso se ha puesto el ejemplo inestable. Si no hay “perturbaciones entre controlador y planta”, en caso estable y fase mínima se puede cancelar lo que se quiera, y se puede conseguir cualquier función de referencia arbitraria M estable (con igual o más grado relativo que G) entre referencia y salida tanto en bucle abierto $u = Qr$, con $Q = MG^{-1}$, como en bucle cerrado (inspirado en control por modelo interno IMC, $u = Ke$, $K = Q/(1 - QG)$) como en cualquier “mezcla” de ambos.

Este ejemplo no tendría sentido con una planta estable y fase mínima sin perturbaciones: incluso, con ruido de medida en sensor, lo óptimo para seguir referencias sería bucle abierto.

[663: c2glhim]	Control con 2 grados de libertad (H-infinito) con referencias y perturbaciones: ejemplo Matlab	****	 14:32
	Materiales: [CÓD.: hinf2glstst.mlx] [PDF]		[YouTube ▶]

En este vídeo se presenta código Matlab para comparar el control para el proceso

$$y(s) = 6.5 \cdot \frac{2 \cdot 3 \cdot 9}{(s-2)(s+3)(s+9)} u(s) + d(s)$$

donde u es la variable manipulada y d es una perturbación (ruido de proceso). La variable y es medible, pero contaminada por un ruido de medida $y_{med} = y + v$, cuyo espectro en frecuencia está acotado por la constante $W_v = 0.01$.

Un caso mas sencillo, sin perturbaciones, se considera en el vídeo [662], cuya visualización previa podría ser aconsejable.


Se plantea la planta generalizada para dos alternativas de control:

- 2 grados de libertad: $u(s) = K_{2GL}(s) \cdot \begin{pmatrix} e(s) \\ r(s) \end{pmatrix}$, ver motivación y definiciones básicas del control 2GL en vídeo [517].
- 1 grado de libertad: $u(s) = K_{1GL}(s) \cdot e(s)$

suponiendo determinada cota de contenido en frecuencia de perturbaciones W_d y limitando el actuador a un ancho de banda de 20 rad/s.

Una vez obtenida la solución \mathcal{H}_∞ a ambos casos, se comparan las respuestas temporales y en frecuencia de dichas soluciones.


En la parte final del vídeo se esbozan los resultados con proceso estable (cambiando $(s-2)$ por $(s+2)$) –las diferencias no son tan grandes en dicho caso–, así como en problemas donde se desee optimizar: (a) seguimiento de referencia sin perturbaciones o (b) cancelación de perturbaciones sin cambios de referencia, en vez de los cambios simultáneos que los diseños iniciales habían considerado.

Nota: El control con 2 grados de libertad es una estrategia que tiene interés práctico incluso en estructuras de control tipo PID, ver Sección 14.7. **** 

Importante: El control óptimo optimiza la respuesta de las salidas controladas (y acción de control) ante las perturbaciones que se han considerado en las entradas de la planta generalizada. No obstante, la respuesta óptima conseguida ante dichas entradas puede resultar en respuestas malas ante otras no consideradas, por lo que se requiere un cierto cuidado en el planteamiento de los problemas de control. En concreto, dependiendo del proceso, los controladores diseñados en los ejercicios anteriores podrían no funcionar: es necesario comprobar sensibilidad a errores de modelado –control robusto, Parte V – y comprobar si el optimizador ha cancelado dinámicas lentas que sean excitadas por perturbaciones no tenidas en cuenta en la minimización de la norma (ver vídeo [mxscan(11:34)]), con lo que podría ser aconsejable replantear el diseño considerándolas (vídeo [mxscan2(08:18)]). ****

☰

19.3.7 Ejemplos: control de sistemas térmicos

[664: termgp]	Construcción de planta generalizada para control multivariable de un sistema térmico de orden 4 (Matlab)	****	 12:54
	Materiales: [CÓD.: termexamplePlantaGen.mlx] [PDF]		[YouTube ▶]

Este vídeo construye una planta generalizada que plantea el control de las temperaturas de dos estancias de un modelo de casa de orden 4. Se abordará dicha construcción utilizando únicamente representación interna, de modo similar a lo desarrollado teóricamente en el vídeo [645].

Detalles teóricos sobre el modelado de este sistema en concreto aparecen en el vídeo [18], el código Matlab y el análisis de propiedades en el vídeo [274], y el dimensionamiento de actuadores (controlabilidad entrada-salida) se discute en los vídeos [439] y [440]. Se aconseja visualizar previamente al menos los dos primeros vídeos referidos al modelado físico del sistema para una mejor comprensión de las manipulaciones que se harán con dicho modelo aquí.

El vídeo plantea un problema de control de temperaturas de las dos habitaciones (1GL, el controlador dispone del error), disponiendo de medidas auxiliares de la temperatura de la buhardilla (seguramente poco útil en la práctica, pero puesta intencionalmente aquí para ilustrar cómo incorporar dichas salidas auxiliares), y disponiendo de una medida de la temperatura ambiente exterior (para poder hacer, si el optimizador lo considera adecuado, algún tipo de prealimentación con dicha perturbación medible).

Se dibuja un diagrama de bloques, se nombran todas las entradas y salidas generalizadas y se calculan las matrices de un modelo en representación interna de la planta generalizada ($A, \hat{B}, \hat{C}, \hat{D}$) a partir de las matrices del modelo físico (A, B, C, D), añadiendo entradas y salidas.

[665: forn6pg] **Caso de estudio horno lineal orden 11: control centralizado, planta generalizada seg. referencias**

Materiales: [[CÓD.:](#) desacsvd3actHORNOh2hinf.mlx] [[PDF](#)]

**** 07:04

[YouTube ▶]

Este vídeo plantea la construcción de una planta generalizada par abordar el control centralizado (en vídeos posteriores: \mathcal{H}_2 en el vídeo [666] y \mathcal{H}_∞ en el vídeo [667]) de un modelo de horno cuyo modelado y control PID por desacoplamiento SVD se abordó en los vídeos [441] y [498].

Se propone construir la planta generalizada de tres formas (desarrollo muy parecido al del vídeo [642]):

- Como una matriz de transferencia (con elementos igual al proceso físico repetidos),
- Como una matriz de transferencia expresada de modo que el proceso físico sólo aparezca una vez en las expresiones,
- Manipulando la representación interna directamente, como el vídeo [645].

Numéricamente, la segunda o, quizás preferiblemente la tercera, serían aconsejables frente a la primera.

[666: forn7h2] **Caso de estudio horno lineal orden 11: control óptimo \mathcal{H}_2**

Materiales: [[CÓD.:](#) desacsvd3actHORNOh2hinf.mlx] [[PDF](#)]

***** 13:59

[YouTube ▶]

En este vídeo se plantea un control óptimo \mathcal{H}_2 para seguimiento de 11 referencias con 3 actuadores de un modelo de horno lineal. El modelado, análisis de propiedades y diseño de PID's por desacoplamiento SVD se detalla en los vídeos [441] y [498]. Supondremos la ganancia estática descompuesta como USV^T .

El detalle de cómo construir la planta generalizada se presenta en el vídeo [665].

En este vídeo se plantean varias posibilidades en cuanto al diseño de los filtros para la planta generalizada ponderada:


- Filtros unidad en error y acción de control, que son razonables (para un primer intento al menos) dado que la planta se había escalado para que ± 1 fuera la amplitud de referencia en entradas y salidas. El resultado es un controlador que "descubre solito" el desacoplamiento SVD, pero con un error de posición que se desea corregir.
- Aumentar la ponderación a baja frecuencia del error (en las dos direcciones más controlables dadas por las dos primeras columnas de U).
- Disminuir la ponderación a baja frecuencia de la acción de control (en las dos direcciones de mayor ganancia, dadas por las dos primeras columnas de V).

Tanto la segunda como la tercera opción consiguen esa "acción integral" para que no haya error elevado ante escalón; todas presentan una respuesta en componentes principales desacopladas, como los PIDs que son la solución base: la primera porque descubre que es la mejor opción, la segunda y la tercera porque aparecen U o V explícitamente en los pesos de la planta generalizada.

Un enfoque similar en \mathcal{H}_∞ de construcción de pesos basado en desacoplamiento SVD se discute en el vídeo [655].

El control \mathcal{H}_∞ del mismo modelo de horno que aquí se aborda en el vídeo [667], que continua el caso de estudio.

[667: forn8hi] **Caso de estudio horno lineal orden 11: control óptimo**
H-infinito, comparación de prestaciones con H2 y PIDs
Materiales: [[CÓD.: desacsvd3actHORNOh2hinf.mlx](#)] [[PDF](#)]

*****  14:51

[YouTube ▶]

Este vídeo plantea el control óptimo \mathcal{H}_∞ para seguimiento de 11 referencias con 3 actuadores de un modelo de horno lineal.

Los dos primeros minutos resumen vídeos anteriores: el modelado, análisis de propiedades y diseño de PIDs por desacoplamiento SVD se detalla en los vídeos [441] y [498]. Supondremos la ganancia estática descompuesta como USV^T . El detalle de cómo construir la planta generalizada se presenta en el vídeo [665]. El control \mathcal{H}_2 se aborda en el vídeo [666].


Los pesos están contruidos teniendo en cuenta el ancho de banda de cada maniobra, con un enfoque similar de construcción de pesos basado en desacoplamiento SVD al que se discute en el vídeo [655].

La parte final del vídeo hace un análisis comparativo de las respuestas en tiempo y frecuencia de los diseños PID, \mathcal{H}_2 y \mathcal{H}_∞ , así como de la robustez [ncfmargin](#) (la teoría de ese margen de robustez se aborda los vídeos [805] y [806]).

La implementación práctica debería, quizás, haber considerado un modelo discreto en el diseño ([675]), o bien discretizando por 'tustin' (por ejemplo), obtener un regulador discreto que implementar, ver ejemplo de código en vídeo [676].

19.3.8 Observadores estacionarios \mathcal{H}_2 y \mathcal{H}_∞

[668: h2obsml] **Observador óptimo \mathcal{H}_2 , equivalencia con Kalman: ejemplo**
Matlab estimación fuerza sistema mecánico
Materiales: [[CÓD.: KalvsH2.mlx](#)] [[PDF](#)]

****  11:31

[YouTube ▶]

Este vídeo plantea una síntesis equivalente alternativa al filtro de Kalman estacionario ([kalman](#), [dlqe](#)), usando el comando [h2syn](#) sobre una planta generalizada adecuada, planteando la estimación del estado como un caso particular del problema generalizado del vídeo [639].

El vídeo utiliza el mismo modelo que el discutido en el caso de estudio de los vídeos [608] y [609].

La planta generalizada se construye directamente en representación interna, según discutido en el vídeo [645].

Se comprueba que `h2syn` y `kalman` producen, con los parámetros adecuados, exactamente el mismo resultado, y se simula el filtro resultante. Este diseño \mathcal{H}_2 será el punto de referencia sobre el que comparar diseños \mathcal{H}_∞ en los vídeos [669] y [670], cuya visualización se aconseja.

Nota: Realmente, la unión de observador de Kalman y realimentación del estado por control óptimo (LQR) puede ser codificada, de forma equivalente, como un problema \mathcal{H}_2 , ver vídeo [647].

[669: hinfobs1] Observador \mathcal{H}_∞ , ejemplo Matlab estimación fuerza (I): pesos constantes

Materiales: [[CÓD.:](#) KalvsHinfiltter.mlx] [[PDF](#)]

**** 09:41

[YouTube ▶]

Este vídeo plantea el diseño de un observador \mathcal{H}_∞ sobre la misma planta generalizada que la considerada en el vídeo [668], donde se vio que el observador \mathcal{H}_2 es el filtro de Kalman estacionario.

Ese diseño \mathcal{H}_2 es el marco de referencia con el cual el observador \mathcal{H}_∞ debe compararse. En este primer vídeo se exploran las propiedades en frecuencia y la simulación temporal del observador \mathcal{H}_∞ con pesos constantes. Se observa que el error “peor caso” –pico en respuesta en frecuencia de perturbaciones a error de observación– es más pequeño que con el filtro de Kalman (obviamente, dado que eso es lo que buscamos minimizar); sin embargo, la amplificación del ruido de medida es muy alta, posiblemente inaceptable en la mayoría de las aplicaciones: que un control sea “óptimo” no significa que sea “bueno en la práctica” si el criterio de optimización no es el adecuado para la situación a enfrentar. Por ello, se requiere usar pesos que varían según la frecuencia, según abordado en el vídeo [670] siguiente.

[670: hinfobs2] Observador \mathcal{H}_∞ , ejemplo Matlab estimación fuerza (II): pesos dependientes de la frecuencia

Materiales: [[CÓD.:](#) KalvsHinfiltter.mlx] [[PDF](#)]

**** 10:54

[YouTube ▶]

Este vídeo plantea el diseño de un observador \mathcal{H}_∞ sobre la misma planta generalizada que la considerada en el vídeo [668], donde se vio que el observador \mathcal{H}_2 es el filtro de Kalman estacionario.

El vídeo anterior [669] planteaba, sobre el mismo proceso, pesos en la planta ponderada constantes, que producían una elevada varianza debido al ruido de alta frecuencia. Este filtro plantea tres posibles diseños alternativos para los pesos:

1. Limitación de altas frecuencias por debajo de cierto filtro paso-bajo (límite por encima de Kalman), y limitación de bajas frecuencias por una constante a minimizar (que quedará por debajo de Kalman).
2. Limitación de altas frecuencias igual que en el caso anterior, pero limitación de bajas frecuencias con una plantilla paso-alto que enfatiza el deseo de un error reducido a muy bajas frecuencias.
3. Combinación de los requisitos de baja frecuencia del segundo caso con una atenuación de las altas frecuencias por debajo de las del filtro de Kalman.

La conclusión es que, dado que el filtro de Kalman es el de mínima varianza (integral del cuadrado de la respuesta en frecuencia), utilizar \mathcal{H}_∞ para bajar el error de estimación a ciertas frecuencias debe ser forzosamente a costa de subir por encima del filtro de Kalman a otras frecuencias (la integral debe ser mayor o igual). Las implicaciones prácticas de ese compromiso (efecto “cama de agua”) se exploran en los diagramas de respuesta en frecuencia y simulaciones temporales de las tres alternativas de diseño arriba enumeradas.


El observador H-infinito tiene orden mayor que la planta “física” observada (tiene el orden de la planta física + los pesos de la planta generalizada ponderada).

19.4 Limitaciones de prestaciones en procesos inestables o de fase no mínima


Esta sección presenta ejemplos de cómo los estimados de anchos de banda y prestaciones alcanzables discutidos de forma preliminar en la Sección 13.2 deben ser revisados cuando existen polos inestables (la inestabilidad requiere una ganancia “mínima” del control para estabilizar y ser especialmente cauto en evitar saturación), y cuando existen ceros en el semiplano derecho (procesos de fase no mínima).

[671: minust]

Acción de control mínima para estabilizar: enfoque H-infinito
(ejemplo Matlab)

***  10:54

Materiales: [[CÓD.:](#) ControlMinimaAccion.mlx] [[PDF](#)]


[YouTube](#) 

En este vídeo se plantea la planta generalizada para calcular el controlador tal que la norma infinito desde perturbaciones (en entrada y salida de la planta) hasta la acción de control sea minimizada. Se comprueba que, como era intuitivamente de esperar, la mínima norma es cero en procesos estables en bucle abierto y distinta de cero en procesos inestables. Si los procesos son simultáneamente *inestables* y de *fase no mínima* entonces la norma de la acción de control crece mucho.


El vídeo reflexiona, basándose en los resultados, en la necesidad de asegurar que los actuadores no saturen ante las perturbaciones que sean previstas cuando el proceso es inestable, dado que al saturar el bucle se abre y, por tanto, la gran mayoría de procesos inestables cuyos actuadores entran en saturación se mueven rápidamente a estados *no recuperables* y se produce un fallo en el sistema. Cuanto mayor sea el mínimo control necesario para estabilizar, menos margen quedará para intentar, además, hacer el error reducido en algún rango de frecuencias; de todos modos, la pérdida de prestaciones en procesos inestables o fase no mínima, así como la sensibilidad posiblemente reducida ante errores de modelado no son objetivo de este vídeo.

[672: limprhpm]

Limitaciones de prestaciones por polos y ceros en semiplano
derecho: ejemplo Matlab

****  12:39

Materiales: [[CÓD.:](#) limitacionesprestacionesV.mlx] [[PDF](#)]

[YouTube](#) 

En este vídeo se comparan las prestaciones temporales y frecuenciales que pueden ser alcanzadas para el proceso $G(s) = \frac{7*40*(2+\beta s)}{(\alpha s+1)(s+4)(s+20)}$ según los valores de $\alpha \in \{-1, +1\}$ y $\beta \in \{-1, +1\}$. El vídeo [671] ya discutió en estos casos la mínima acción de control necesaria para estabilizar y su influencia en el dimensionamiento del proceso.

Todos los procesos tienen el mismo diagrama de Bode de amplitud, pero el desfase cambia; se prueban los casos estable+fase mínima, inestable+fase mínima, estable+fase no mínima, inestable+fase no mínima. Se diseña un controlador \mathcal{H}_∞ que trate de minimizar el error en un ancho de banda lo más grande posible sujeto a restricciones de amplitud de actuador y de picos de la función de sensibilidad.

Como conclusión, el caso de fase no mínima reduce mucho el ancho de banda alcanzable, y se reduce mucho más si el proceso es simultáneamente inestable y de fase no mínima; en este caso el ancho de banda es muy bajo y los picos de las distintas funciones de bucle cerrado son altos lo que dará lugar a prestaciones malas y poca tolerancia a errores de modelado. Este vídeo es la reinterpretación en \mathcal{H}_∞ de las ideas preliminares sobre limitación de prestaciones en el control de procesos de fase no mínima que aparecían en el vídeo [398] en un contexto de control por modelo interno.

19.5 Antiwindup en controladores en representación interna

[673: awsste]
Anti-windup en reguladores genéricos lineales en representación interna

12:54

Materiales: [AWTrackStateSpace.pdf]

[YouTube ▶]

Este vídeo presenta las ecuaciones que posibilitan convertir cualquier regulador lineal en representación interna a una representación con "antiwindup", mediante la adición de un término de corrección $L_a \cdot (u_{real} - u_{ideal})$ a la ecuación de estado. La parte inicial del mismo revisa la motivación hacia el antiwindup (polos inestables en reguladores) y las fórmulas para PID tracking-mode (vídeo [521]), IMC (vídeo [417]) u observador (alimentar u_{real} en vez de u_{ideal} al observador, ver ejemplo numérico en vídeo [570]).

El caso PID tracking-mode y observador son casos particulares de esta metodología, como se discute en el vídeo [674].

Ejemplos de simulación de las metodologías discutidas en este vídeo se abordan en el vídeo [581] (caso particular observador) y en el vídeo [676] (caso genérico de un regulador \mathcal{H}_∞ donde el antiwindup se calcula por asignación de polos como se ha propuesto en el presente material).

[674: awsscp]
Anti-windup en reguladores genéricos lineales en representación interna (2): casos particulares PID y observador

11:22

Materiales: [AWTrackStateSpace.pdf]

[YouTube ▶]

Este vídeo es continuación del vídeo [673], donde se describía cómo convertir un regulador lineal en representación interna a un esquema antiwindup o bumpless transfer, mediante la adición de un término corrector $L_a \cdot (u_{real} - u_{ideal})$ a la ecuación de estado. Las ideas principales de dicho vídeo son repasadas rápidamente al inicio de éste.


Este vídeo comprueba que cuando la representación estudiada es la de un PI con feed-forward, el esquema general acaba replicando las ecuaciones de un PI con tracking-mode antiwindup (los vídeos [521] y [522] lo estudian en detalle el caso PID sin introducir la representación interna, esto es, en FdT).

También el antiwindup en observadores (alimentar u_{real} al proceso, ejemplo Matlab en vídeo [570]) es un caso particular del esquema general.

Un ejemplo detallada de la implementación antiwindup "explícita" como observador y de la implementación genérica como regulador en representación interna y de la equivalencia de ambas opciones de código se aborda en el vídeo [581]. Como se ha demostrado teóricamente aquí, en dicho vídeo la simulación es idéntica se gaste la implementación que se gaste.

19.5.1 Caso de estudio: implementación en un regulador \mathcal{H}_∞ discreto

[675: hiimpl1] Control H-infinito de proceso discreto 2o orden (seguimiento referencias + rechazo perturbación a la entrada)
Materiales: [[CÓD.:](#) AWSStestHinfsyn2.mlx] [[PDF](#)]


****  08:47
 [YouTube ▶]

En este vídeo se hace un diseño \mathcal{H}_∞ en tiempo discreto para un proceso de segundo orden (el mismo del caso del vídeo [575] con observador y realimentación del estado, si deseáis comparar).

Se plantea una planta generalizada para un problema de seguimiento de referencias y, simultáneamente, cancelación de perturbaciones a la entrada. Se discuten los significados de los filtros, y se presenta el diseño final con norma infinito (ponderada) igual a 1, y se analizan las prestaciones en bucle cerrado (con [lft](#)) en tiempo y en frecuencia.

La simulación e implementación de este controlador incorporando *antiwindup* se discute en el vídeo [676], continuación de éste.

[676: hiimpl2] Control H-infinito de proceso discreto 2o orden (seguimiento referencias + rechazo perturbación a la entrada): implementación y antiwindup
Materiales: [[CÓD.:](#) AWSStestHinfsyn2.mlx] [[PDF](#)]

***  14:49
 [YouTube ▶]

En este vídeo se simula (bucle “for”) un diseño \mathcal{H}_∞ en tiempo discreto para un proceso de segundo orden, para un problema de seguimiento de referencias y, simultáneamente, cancelación de perturbaciones a la entrada. Aunque aquí se resume rápidamente el diseño en los dos primeros minutos, la explicación detallada del mismo se aborda en el vídeo [675].


En este vídeo, nos concentramos en el código de simulación.

Como el resultado de [hinfsyn](#) es un regulador en representación interna, la simulación se hace en dos líneas: ecuación de salida para calcular u_k y luego la ecuación de estado para calcular el estado x_{k+1}^c del controlador para la siguiente muestra. Esto es muy similar a la implementación “ABCD” genérica del regulador con observador discutido en el vídeo [581]: TODOS los controladores lineales discretos en representación interna se simulan igual.

El código se completa con refinamientos de paso a incrementales para un punto de funcionamiento no nulo (unidades físicas absolutas), así como con la incorporación de antiwindup; en este caso hemos calculado la ganancia de antiwindup mediante asignación de polos (teoría en vídeo [673]), cosa que no era necesaria en los controles basados en observador en el vídeo antes referido.

19.6 Conclusiones

[677: icscp1c] Conclusiones/resumen control sistemas complejos (dimensionamiento, estructuras PID, control generalizado)
Materiales: [ConclusionesICSCP1.pdf]

****  17:57
 [YouTube ▶]

Este vídeo resume las ideas principales de los vídeos anteriores sobre selección de variables controladas y manipuladas (dimensionamiento de actuadores), sobre estructuras de control y sobre el problema de control generalizado. Realmente corresponde al primer parcial de una de mis asignaturas de máster que impartí, pero lo publico por si estas conclusiones/resumen de material anterior resultan de utilidad para asentar ideas antes de abordar control robusto, no lineal, planificación de ganancia, etc.

Parte IV

Análisis multivariante, Identificación y Detección de Cambios

Capítulo 20


Identificación/ajuste de modelos (sin dinámica)

Nota: Los contenidos de este capítulo requieren una revisión en profundidad de los conceptos básicos de estadística (Sección B.2) y de predicción lineal óptima mínimo-cuadrática, en la Sección B.5.

20.1 Mínimos cuadrados (caso estático)

[678: id1a] Identificación, generalidades: modelos y su uso (simulación, observación, identificación paramétrica)

Materiales: [IdentEstat1.pdf]

***  04:34


[PLAY ►]

Este vídeo discute generalidades sobre lo que es un modelo matemático determinista o aleatorio, así como su posible uso en problemas de simulación, observación, identificación paramétrica... discutiendo sobre posibles definiciones sobre qué se busca, esto es, si una estimación puntual versus probabilística (funciones de densidad, intervalos de confianza).

La parte final del vídeo define los modelos *paramétricos* (estáticos): una función de densidad condicional $f(y|x, \theta)$, o modelos deterministas $y = f(x, \theta)$.

[679: id1b] Identificación mínimos cuadrados lineal: propiedades estadísticas del estimado, matriz de información (Fisher)


Materiales: [IdentEstat1.pdf]

***  19:54

[PLAY ►]


Dado un modelo matemático $f(y|\theta)$, esto es, una probabilidad (condicional) de y dado θ , se plantean tres tipos de definiciones de lo que podría ser la *identificación* de θ : puntual máxima probabilidad (moda), máxima probabilidad a posteriori (Bayes), o estimación bayesiana completa de función de probabilidad a posteriori.

Luego se abordan los mínimos cuadrados lineales:

- Identificación en ecuación lineal $y = X\theta + v$, $v \sim \mathcal{N}(0, \sigma)$.
 - Equivalencia entre predicción de “máxima probabilidad” y “mínimos cuadrados” deterministas, y también con mejor predicción lineal (error de predicción no correlado con datos X). La mejor predicción lineal está discutida en el vídeo [966] y [967], y las tres interpretaciones [determinista (optimización) / geométrica (proyección) / estadística (mínima varianza)] de la fórmula de mínimos cuadrados están detalladas en el vídeo [971] ().
 - Mínimos cuadrados: pseudoinversa.
 - matriz de varianzas-covarianzas de parámetros estimados.
 - matriz de información (Fisher), componentes principales de X y su interpretación. Identificabilidad.
 - Diseño de experimentos: qué hacer si hay poca información. Tomar más datos, rediseñar experimento, simplificar el modelo (regularización). Véase un ejemplo de regularización en vídeo [737].
 - Si la información es buena y ajuste malo: o bien modelo es bueno (pero ruido aleatorio grande), o bien el modelo no es correcto.
 - Validación (*training* versus *test* data set).
 - Compromiso entre error sistemático y variabilidad (*bias/variance* tradeoff). Sobreparametrización.
- *Como curiosidad, estimar $y \approx 2x$ con mínimos cuadrados no implica que estimar el modelo inverso vaya a resultar $x \approx 0.5y$, ver vídeo [969].

[680: optimML] Determinación de parámetros de un modelo probabilístico: ejemplo Matlab (max likelihood)


Materiales: [CÓD.: MLVza.mlx] [PDF]

***  07:24

[YouTube ►]

Este vídeo presenta un caso sencillo de estimar un parámetro de un modelo probabilístico a partir de realizaciones experimentales maximizando la probabilidad (bueno, en rigor, la función de densidad) de los datos experimentales, esto es, el estimado max-likelihood.

El ejemplo es muy sencillo: dadas tres muestras de una distribución normal de media cero, estimar el parámetro de desviación típica σ de la misma. Si la densidad es $f(y, \sigma)$ y las muestras son independientes, se trata de maximizar $J = f(y_1, \sigma)f(y_2, \sigma)f(y_3, \sigma)$. La maximización en sí se hace por “fuerza bruta”, comprobando un montón de valores a ver cuál da un valor más alto de J . Se comprueba numéricamente que el óptimo coincide con la estimación de la desviación típica muestral $\sqrt{(y_1^2 + y_2^2 + y_3^2)/3}$.


[681: id2] **Identificación (mínimos cuadrados, sistemas estáticos):** *** / ****  20:51
MATLAB
Materiales: [[CÓD.:](#) IdentStaticSISOvideoid2.mlx] [[PDF](#)] [PLAY ►]

Este vídeo analiza (Matlab) los conceptos de ajuste por mínimos cuadrados, experimento informativo, combinación de parámetros (dirección principal) más difícil de identificar.

También se calcula la desviación típica del error de predicción, y la matriz de varianzas-covarianzas de los parámetros estimados.

[09:07] Se descompone la variabilidad del error de predicción en (i) variabilidad que podría ser reducida si se tuviera un experimento más informativo y (ii) variabilidad que el modelo considera que debe ser ruido blanco aleatorio. Con ello, puede determinarse la pertinencia de tomar o no más datos experimentales. ****


[13:28] Con la diagonal de la matriz de varianzas-covarianzas de los parámetros estimados se obtiene la desviación típica de cada parámetro, y con ella un intervalo de confianza (suponiendo ruido de distribución normal) en la estimación de parámetros. Como las suposiciones de la teoría no siempre se cumplen “suficientemente bien” en los experimentos, hay que interpretar con mucha precaución los resultados sobre varianza de parámetros estimados. El vídeo [748] argumenta sobre ideas similares en un caso de estimación de modelos de primer y segundo orden con retardo.

[682: id3] **Ajuste de modelos no lineales estáticos** ****  22:06
Materiales: [id3.zip] [PLAY ►]

Este vídeo discute el problema de ajuste de modelos estáticos no lineales:

- Modelos no lineales, planteamiento del problema.
- Caso particular: modelos lineales en parámetros $y = \theta_1 \sin(x) + \theta_2 x^2 + \nu$, $\nu \sim \mathcal{N}(0, \sigma)$.
- Caso no lineal: optimización genérica. Matriz de información a partir de la linealización en los parámetros resultantes.
- Ejemplo Matlab de regresión no lineal, usando `fitnlm` y `predict`. Validación del residuo con `plotResiduals`.


Caso de estudio (académico)

[683: rlce1] **Regresión lineal (caso estudio Matlab): centrado, escalado, pseudoinversa, modelo de predicción** **  09:15
Materiales: [[CÓD.:](#) rlce123.mlx] [[PDF](#)] [YouTube ►]

Este vídeo presenta un caso de estudio de ajuste de un modelo para predecir una variable y a partir de tres variables x .

Aquí se discuten los preliminares: el preprocesado de variables (centrado y escalado), la fórmula de la pseudoinversa, y el modelo de predicción en unidades originales no centradas.


El análisis del error y la varianza de los parámetros estimados se aborda en el vídeo [684], continuación de éste.

[684: rlce2]	Regresión lineal (caso estudio Matlab): error de predicción, varianza de parámetros estimados Materiales: [CÓD.: rlce123.mlx] [PDF]	****  13:28 [YouTube ▶]
--------------	--	---

Este vídeo continua el caso de estudio del vídeo [683]. Se analiza la varianza explicada por el modelo y la matriz de varianzas-covarianzas de los parámetros estimados, obteniendo intervalos de confianza $\pm 2\sigma$ (intervalos que deben ser considerados como aproximaciones “muy informales”, dado que incluso ante modelos generados por un proceso lineal con ruido gaussiano deben utilizarse fórmulas más sofisticadas).

Se compara una serie de datos (caso 1) que da un modelo básicamente inútil, con otra serie de datos (caso 2) que da un modelo muy bueno.


El caso 1 produce una gran incertidumbre en los parámetros estimados, considerando su distribución marginal (independientemente del resto); el elipsoide de confianza daría la descripción más exacta, y motiva la regresión por componentes principales que aparece en el vídeo [685].

[685: rlce3]	Regresión lineal (caso estudio Matlab): regresión por componentes principales PCR, simplificación de modelo Materiales: [CÓD.: rlce123.mlx] [PDF]	*****  14:25 [YouTube ▶]
--------------	--	--

Este vídeo aborda la regresión por componentes principales en el caso estudiado en el vídeo [684]; aunque en las variables originales todos los parámetros tenían un intervalo de confianza (marginal) grande, el elipsoide, alineado con los ejes principales permite dar un intervalo que no incluye el cero para un parámetro. Hacer regresión lineal estimando ese único parámetro es lo que se denomina regresión por componentes principales (PCR), como se discute en este vídeo. El enfoque teórico formal a la PCR se aborda en el vídeo [718], que se recomienda visualizar, y otro ejemplo Matlab se discute en el vídeo [719].

20.2 Mínimos cuadrados recursivos

20.2.1 Algoritmos recursivos sin olvido

[686: mcr1]	Mínimos cuadrados recursivos 1 (teoría, sin olvido) Materiales: [mcr1.pdf]	*** / ****  11:18 [YouTube ▶]
-------------	---	---

– Motivación y planteamiento del problema: detener experimentos cuando la información sea suficiente, control adaptativo, monitorización o detección de fallos.

– Mínimos cuadrados recursivos: a partir de una estimación a priori $\theta \sim \mathcal{N}(\hat{\theta}_0, \Sigma_{\theta,0})$ y datos X en un modelo (estático) $y = X\theta + v$, obtener $\hat{\theta}_1$.

– Uso de la fórmula de mejor predicción lineal (vídeo [967]) para resolver el problema.

– Relación con el filtro de Kalman.

La modificación a los algoritmos para incluir un “olvido” capaz de seguir parámetros cambiantes se discute en el vídeo [689].

[687: mcr1m] Mínimos cuadrados recursivos sin olvido: ejemplo Matlab
Materiales: [[CÓD.:](#) recursiveteststaticSinOlvido.mlx] [[PDF](#)]

***  05:41
 [YouTube ►]

Este vídeo implementa dentro de un bucle de simulación el código Matlab con las fórmulas demostradas en el vídeo [686], para identificar un modelo de un único parámetro $y = x\theta + v$, con $v \sim \mathcal{N}(0, 1)$.

Se presenta la evolución de los estimados en media y desviación típica (intervalos de confianza) conforme se van incorporando nuevos datos para la identificación.

[688: mcrnr] Mínimos cuadrados recursivos: relación con fórmula no recursiva (pseudoinversa)

Materiales: [mcrvscclasico.pdf]

*****  07:43

[YouTube ►]

Este vídeo demuestra que los mínimos cuadrados recursivos (teoría en vídeo [686]), cuando la incertidumbre inicial (matriz de varianzas-covarianzas de los parámetros antes de incorporar la primera muestra) tiende a infinito, resultan en una fórmula equivalente a la de la pseudoinversa, utilizando el lema de inversión de matrices (identidad de Woodbury).

Ejemplo adicional (relación con “regularización”): la regularización en un problema de mínimos cuadrados (minimizar $\|y_{N \times 1} - X_{N \times m} \theta_{m \times 1}\|^2 + \lambda \|\theta\|^2$), cuya solución es $\hat{\theta} = (X^T X + \lambda I)^{-1} X^T \cdot y$ (equivalentemente, mediante lema inversión matrices, $\hat{\theta} = X^T (X X^T + \lambda I)^{-1} \cdot y$) puede ser entendida como un problema de mínimos cuadrados recursivos.

En efecto, si el estimado a priori de θ es $\hat{\theta}_0 \sim \mathcal{N}(0, \gamma I)$, y el ruido de medida es $V = \eta I$, entonces se tiene, con el mismo lema de inversión de matrices que en los materiales del vídeo [mcrnr(07:43)], que:

$$\hat{\theta}_1 = \gamma X^T (\gamma X X^T + \eta I)^{-1} y = (X^T X + \frac{\eta}{\gamma} I)^{-1} X^T y$$

con lo que el parámetro de la regularización sería $\lambda = \eta/\gamma$.

20.2.2 Algoritmos con olvido (factor de olvido / filtro de Kalman)

[689: mcr2] Mínimos cuadrados recursivos (II): incorporación de olvido (factor exponencial, filtro Kalman)

Materiales: [mcr2.pdf]

****  10:51

[YouTube ►]

- Motivación: identificación de parámetros variantes (lentamente) en el tiempo.
- Revisión de la fórmula sin olvido (discutida en detalle en el vídeo [686]).
- Modificación de las fórmulas sin olvido: (i) ventana fija, (ii) factor de olvido exponencial, (iii) filtro de Kalman con dinámica de parámetros.

[690: mcr2m] Mínimos cuadrados recursivos con olvido: ejemplo Matlab
Materiales: [[CÓD.:](#) recursiveteststaticCompletoOA.mlx] [[PDF](#)]

****  14:23
 [YouTube ►]

Este vídeo introduce las modificaciones necesarias para incorporar olvido al código Matlab del problema de identificación discutido en el vídeo [687].

– Para demostrar el funcionamiento del olvido, la ganancia a estimar no es constante, sino que se le hace cambiar siguiendo una onda cuadrada (fallo repentino, por ejemplo).

- Se programa el olvido exponencial, y se comprueba la evolución en media e intervalo de confianza del parámetro estimado.
- También se programa la versión filtro de Kalman, para comparar los resultados con el caso anterior.
- Se simula qué ocurre cuando los nuevos datos no son informativos: aumenta la incertidumbre en el parámetro estimado.

Nota: Podría resultar de interés en este momento el comparar el código de los mínimos cuadrados recursivos con el del filtro de Kalman del vídeo [kalml(33:33)], comprobando el paralelismo entre ambos.

[691: mcrsup] Mínimos cuadrados recursivos: supervisión del factor de olvido *****  16:30
Materiales: [MinimosCuadradosRecursivosSuperv.pdf] 

- Motivación: parámetros cambiantes, pero posibles fragmentos de datos no excitantes requieren supervisar el olvido.
- Revisión de los conceptos fundamentales del vídeo [689] y simulaciones como las del vídeo [690].
- Supervisión del olvido en filtro de Kalman: parámetros incrementales alrededor de unos valores nominales, dinámica asintóticamente estable.
- Supervisión del factor de olvido exponencial: no olvidar si el máximo valor propio o la traza de la matriz de varianzas-covarianzas está por encima de un cierto umbral.
- Compromisos de diseño: estabilidad numérica versus retrasos en la detección de cambios súbitos o desviaciones en los parámetros estimados.

20.3 Mínimos cuadrados totales

[692: t1s1] Mínimos cuadrados totales TLS: teoría *****  19:15
Materiales: [solotls.pdf] 

Este vídeo discute el problema de mínimos cuadrados “totales” (*total least squares*, TLS, en la literatura en lengua inglesa) que resuelve el ajuste de un modelo $y = C \cdot x$, con medidas de x , y pero considerando que ambas están corrompidas por ruido de desviaciones típicas σ_x , σ_y respectivamente. En ese caso, los mínimos cuadrados standard, esto es, $\hat{C} = \text{pinv}(X)y$, son un estimador *sesgado*, esto es, con un error incluso en media (error sistemático).

La propuesta para resolver el TLS y obtener un estimado de C no sesgado es formar la matriz de varianzas covarianzas de (x, y) y restarle a dicha matriz una diagonal $\text{blkdiag}(\sigma_x^2 I, \sigma_y^2 I)$.

*Si se hace escalado/normalizado de los datos para que la desviación típica del ruido de medida de x e y sean idéntico, entonces formando la matriz de datos con todas las muestras de (x, y) , y computando su SVD, el modelo TLS se asocia a las direcciones (filas de U) asociadas a los *valores singulares más pequeños*. Esto está directamente relacionado con lo que se denomina análisis de componentes principales, discutido en otros vídeos de la presente colección.

Ejemplos Matlab de TLS aparecen en los vídeos [693] y [694].

20.3.1 Ejemplos Matlab

[693: t1sm] Mínimos cuadrados totales TLS: ejemplo Matlab (estático)

**** 12:37

Materiales: [[CÓD.:](#) t1spruebaGANb.mlx] [[PDF](#)]

[YouTube ▶]

En este vídeo se ilustra, mediante cálculos con Matlab, la teoría del vídeo [692] con un ejemplo de estimación por mínimos cuadrados “clásicos” versus “totales” de la constante k de un muelle con ecuaciones $F = k \cdot x$, supuestas disponibles un total de N medidas de las señales F y de x , ambas con ruido de medida significativo.

Primero se comprueba que, en efecto, los mínimos cuadrados clásicos (pseudoinversa) producen un estimado sesgado de la ganancia del muelle. Este sesgo está, en cierto modo, relacionado con la idea de que el modelo inverso estadístico no coincide con el modelo algebraico (vídeo [969]).

A continuación, se forma la matriz de datos con todas las muestras de (F, x) , de dimensiones $2 \times N$, escaladas a la misma varianza del ruido de medida. Se le hace el SVD y se obtienen dos componentes principales de varianzas 5.8 y 0.34. La columna de U asociada al componente de baja varianza estima, como estaba previsto, la constante “real” del muelle mejor que la fórmula usual de pseudoinversa (una vez deshecho el cambio de variable que escalaba a la misma varianza de ruido de medida, obviamente).

Nota: la mejor predicción de fuerza dada x con ruido sigue siendo el estimado de mínimos cuadrados, aunque no coincida con el valor “real” de la constante del muelle “interno”. Estimar el modelo sin ruido que ha originado los datos no es lo mismo que predecir la salida ante los datos de posición ruidosos; el parámetro óptimo es diferente en cada caso.

Un ejemplo con 5 variables aparece en el vídeo [694].

[694: t1s51] Ejemplo mínimos cuadrados totales TLS 5 variables (Matlab)

*** 15:35

Materiales: [[CÓD.:](#) TLS3x2p4.mlx] [[PDF](#)]

[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo presenta un ejemplo de mínimos cuadrados totales (Total Least Squares, TLS) con cinco variables relacionadas entre sí. La teoría se discute en el vídeo [692], y un ejemplo más sencillo (constante de un muelle) se aborda en el vídeo [693].

Cuando se generan los datos, existe un modelo $y_{1 \times 2} = x_{1 \times 3} \Theta_{3 \times 2}$, esto es, de dos ecuaciones (predecimos dos variables a partir de otras tres). Esto no es conocido para el TLS, obviamente, dado que debe averiguarlo.

Es necesario TLS porque todos los datos, tanto de X como de Y están contaminados con ruido de unas ciertas desviaciones típicas; en efecto el estimado de mínimos cuadrados ordinarios con pseudoinversa resulta sesgado. Escalando para que todas las variables tengan ruido 1, los dos valores singulares más pequeños de la matriz $[Y \ X] = USV^T$ son, en efecto, aproximadamente 1 como predice la teoría. Las dos últimas columnas de V constituyen el modelo implícito. Para comprobar que es correcto, se deshace el escalado y se despeja $Y = X \Theta_{TLS}$ para comprobar que coincide con el usado al generar los datos (que, en una aplicación real, sería desconocido).

El vídeo [695], continuación de éste, plantea qué ocurre si me equivoco en el escalado: desgraciadamente, también se obtiene un modelo sesgado en ese caso.

[695: t1s52] Ejemplo mínimos cuadrados totales TLS 5 variables (Matlab): sesgo con escalado incorrecto

**** 11:53

Materiales: [[CÓD.:](#) TLS3x2p4.mlx] [[PDF](#)]

[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo es continuación del vídeo [694] donde se hacía un análisis por mínimos cuadrados totales de las ecuaciones implícitas que relacionaban cinco variables (aleatorias).

Aquí se abordan dos cuestiones. Una primera cuestión, bastante trivial, de cómo modificar el código si cada muestra de los datos es una columna en vez de una fila. Las modificaciones son meramente cosméticas, transponiendo cosas, y el resultado es, obviamente idéntico.


La segunda cuestión, más interesante, aborda analizar qué ocurre cuando el escalado previo al SVD es incorrecto. Desafortunadamente, el resultado es un modelo sesgado. De hecho, el sesgo es mayor incluso que en el modelo de mínimos cuadrados clásicos que es el que se suele usar para justificar TLS... si el escalado está mal diseñado, puede ser peor el remedio que la enfermedad. Esto puede ser un problema en aplicaciones de PCA/TLS.

Dado que el modelo es multivariable, e implícito (ecuación entre cinco variables $\Theta[Y; X] = 0$, se usa el comando `subspace` para determinar el ángulo entre los subespacios en los que las variables $[Y; X]$ pueden estar si verifican el modelo (el espacio nulo de Θ). Así, la magnitud del "sesgo" se reduce a un único número.

20.4 Otros índices de coste para regresión

[696: reglad]

Otros índices de coste para regresión: valor absoluto de error/parámetros (LP;QP)
Materiales: [LeastAbsoluteDeviation.pdf]

***  11:49
[YouTube ▶]

Este vídeo presenta alternativas basadas en el mínimo "valor absoluto" tanto del error de predicción $y - \theta\phi(x)$ de un modelo como de los parámetros (para regularización). Presentan una mayor robustez a "outliers" (valores anómalos, totalmente fuera de modelo) y suelen producir un número de parámetros (si se usan en regularización) con más elementos igual a cero que la regularización $c\|\theta\|^2$.


El inconveniente es que no existe una fórmula "explícita", cerrada, con la solución (en los mínimos cuadrados sí existe, con inversas y pseudoinversas); los parámetros óptimos deben ser calculados numéricamente con programación lineal o cuadrática.

En algunas transparencias en el vídeo aparece q_i en vez de θ_i . Está corregido en el PDF de materiales.

20.5 Métodos Kernel


[697: kermot]

Problemas con gran número de variables por muestra: motivación al enfoque kernel
Materiales: [KernelTrick.pdf]

***  12:15
[YouTube ▶]

En este vídeo se motiva el por qué alguna literatura de *Machine Learning* utiliza con profusión lo que se denominan técnicas "Kernel". En concreto, estas técnicas están indicadas cuando se dispone de pocas muestras (ejemplos de entrenamiento para ajustar un modelo) pero cada muestra contiene muchas "características" (features) que son bien medidas o bien calculadas a partir de medidas. Por ejemplo, podríamos tener que entrenar un sistema de reconocimiento de imágenes a partir de 50 ejemplos, teniendo cada ejemplo 1000 pixels de datos; podríamos tener información sobre 100 ejecuciones de una reacción química industrial (20 minutos de duración, 2000 medidas) y queremos sacar conclusiones sobre qué influye en la calidad del producto...

Este vídeo discute todas esas situaciones y esboza la idea principal (un "cambio de variable") que se detallará en el vídeo [698].

[698: ktrick] **Truco del Kernel (1): eliminación de espacio nulo, matriz de productos escalares** ****  12:27

Materiales: [KernelTrick.pdf]

[YouTube ▶]

Este vídeo revisa someramente las ideas del vídeo [697] y expone el detalle técnico del problema: un problema de optimización $J(\theta) = L(\theta X) + c\|\theta\|$ puede plantearse de forma equivalente en términos de unas variables w , con el cambio $\theta = wX^T$. El número de elementos de w es igual al número de muestras de entrenamiento. Con ello, $J(\theta) = J(w) = L(wK) + c \cdot (wKw^T)$ siendo $K = X^T X$ denominada matriz “kernel”. La elección de regresores en muchos problemas se cambia por la elección de K de entre diversas opciones analizadas en la literatura.

[699: ktrick2] **Truco del Kernel (2): selección de Kernel y construcción de modelo de regresión final** ****  12:53


Materiales: [KernelTrick2.pdf]

[YouTube ▶]

Este vídeo es continuación del vídeo [698]. Presenta distintos ejemplos de Kernel, en concreto el lineal, cuadrático, polinomial de grado arbitrario y el kernel RBF (gaussiano). En la segunda parte del vídeo, presenta cómo reconstruir un modelo de regresión una vez calculados los parámetros “ w ” óptimos (uno por dato). No es necesario “recuperar” el vector θ original (de hecho, si se ha usado una cierta K de la literatura, puede no ser posible), y puede expresarse un modelo de regresión a partir de w . El modelo de regresión resultante tiene una interpretación con cierto “parecido” a las interpolaciones, dado que cada muestra de entrenamiento es responsable de un elemento de w .

Esta es la idea detras de Kernel regression, Support Vector Machines, etc. que este material no aborda en profundidad por brevedad, pero que son técnicas de importancia en el *machine learning* introducidas en los años 1990.


20.5.1 Regresión mínimos cuadrados contraída (ridge-regression), versión Kernel

[700: kerls] **Métodos Kernel: ajuste (regresión) por mínimos cuadrados** ****  08:55

Materiales: [KernelRidgeLSRegression.pdf]

[YouTube ▶]

Este vídeo presenta la reformulación (equivalente) de un problema de regresión de mínimos cuadrados regularizado (ridge-regression) en “versión Kernel” (pocas muestras, mucha info por muestra). El óptimo, como en todos los problemas de mínimos cuadrados se obtiene derivando e igualando a cero una expresión matricial (de forma análoga a la discutida en el vídeo [928]).

[701: kerlsm] **Métodos Kernel: ajuste (regresión) por mínimos cuadrados, Ejemplo Matlab** ****  14:51

Materiales: [[CÓD.:](#) KernelRidgeRegressionTest.mlx] [[PDF](#)]

[YouTube ▶]

Este vídeo presenta un ejemplo Matlab de regresión por mínimos cuadrados, según la teoría vista en el vídeo [700]. Se utilizan los kernels lineal, polinomial y RBF presentados en el vídeo [699].

[702: kstat]

Interpretación estadística de la regresión Kernel mínimos cuadrados, procesos gaussianos

Materiales: [KernelRegressionStatisticInterp.pdf]

En este material se presenta la interpretación estadística como mejor predicción lineal (mínimos cuadrados) de la regresión discutida en el vídeo [700], identificando las matrices del resultado final con varianzas-covarianzas. En efecto, del mismo modo que la regresión “determinista” en su versión clásica no-kernel tenía una interpretación estadística, también la tiene su versión Kernel.

El resultado (suponiendo distribución normal de las variables aleatorias intervinientes) es lo que se conoce como regresión “procesos gaussianos”: un conjunto infinito de variables aleatorias $f(x)$, uno para cada x del dominio donde se dispone información, medidas con ruido $y = f(x) + v$, cuya covarianza o correlación es $\kappa(x, y)$. La función que genera el Kernel pasa a ser entendida como una función “generadora de covarianza”. Esto, en el caso unidimensional tiene una interpretación de “filtrado” en sistemas dinámicos: si $\kappa(x, y) = \tilde{\kappa}(\|x - y\|)$ la distancia significa exactamente lo mismo que la distancia “en tiempo” en un problema de filtrado cuya autocorrelación sea $\tilde{\kappa}$. La transformada de Fourier de $\tilde{\kappa}$ sería el power-spectral-density (según descrito en el vídeo [207]) del filtro asociado al Kernel.

Nota: El siguiente ejemplo Matlab está muy relacionado con el ejemplo explicado en el vídeo [krig2dm(09:25)].

[703: gaussp]

Regresión Kernel: interpretación estadística/filtro de Wiener (Matlab)

Materiales: [[CÓD.:](#) KernelRegresGaussProc.mlx] [[PDF](#)]

Este es un fichero de Matlab que ejemplifica cómo generar una función $\kappa(x, y)$ para Kernel regression en el sentido estadístico descrito en el material [702].

Si se ejecuta la prueba “0”, se puede (comentar/descomentar) ver el efecto de usar un filtro de primer orden $a/(s+b)$ o un filtro de Butterworth de segundo orden $a/(s^2 + \sqrt{2} \cdot \omega s + \omega^2)$ para generar sus power-spectral-density y autocorrelación asociadas, siendo la última la propuesta para κ . La relación con la fórmula de reconstrucción de señales muestreadas de ancho de banda finito también es explorada (teorema de Shannon-Nyquist) haciendo **prueba=1**. En resumen, back to the basics: los métodos Kernel pueden ser reinterpretados como la versión multidimensional de resultados clásicos de “filtro de Wiener”, “Shannon”, etc. en procesamiento de señal.

[704: optimkern]

Optimización de hiperparámetros de un kernel (covarianza proc. estocástico) para regresión

12:57

Materiales: [[CÓD.:](#) OptimHiperparametros.zip] [[PDF](#)]

[YouTube]

En este vídeo se determinan los (hiper)parámetros óptimos de un Kernel gaussiano $Me^{-\frac{x^2}{\sigma^2}}$ para ajustar una serie de datos.

Se utilizan dos criterios: – ajuste a unos datos de validación (error cuadrático, sin ponderar según vza., por simplicidad) – estimado maximum-likelihood (maximizar la probabilidad a priori de los datos experimentales). Una versión más sencilla de la idea se discute en el vídeo [680] que podría ser aconsejable visionar antes de éste.

Se comparan los resultados entre sí y con ajustes “manuales” de dichos parámetros.

[705: kcentra]

Centrado de Kernels

Materiales: [KernelCentering.pdf]

10:59 [YouTube]

En regresión, PCA, etc. se suponen datos de media cero. Al reemplazar los datos en bruto x por un vector de características $\phi(x)$ entonces $[\phi(x_1) \dots \phi(x_N)]$ ya no tiene por qué conservar la media cero, dado que las no-linealidades usualmente cambian la media. Para que ello no influya en los resultados, se debe “centrar” de nuevo ϕ alrededor de su media muestral. Afortunadamente, no es necesario operar sobre ϕ (que puede ser desconocido en forma explícita): operando con la matriz K (Kernel original) y una matriz de tamaño $N \times N$ con todos sus elementos igual a $1/N$ puede construirse un “Kernel centrado”.

Nota: En los ejemplos de código anteriores, por simplicidad no se ha usado el Kernel centrado ni trasladar a media muestral cero los datos. Posiblemente deberían refinarse añadiendo dichos aspectos, o considerando modelos semiparamétricos (con media no nula, *ordinary kriging*), como se hace en la siguiente sección.

20.5.2 Regresión semiparamétrica (kernel+modelo con parámetros explícitos)

[706: semipar1] Modelos semiparamétricos (Kernel+regresores): estimación de salidas y parámetros ****  07:53

Materiales: [SemiParametricRegression.pdf] [YouTube ▶]

Este vídeo plantea la regresión en modelos $y(x) = z(x) + \phi(x)\theta + \epsilon$ siendo ϵ un ruido de medida, θ un vector de parámetros ajustables, $\phi(x)$ un vector de regresores, y $z(x)$ un componente de tipo “proceso Gaussiano” con matriz de varianzas-covarianzas $\kappa(x_1, x_2)$ conocida.

El resultado es equivalente a utilizar en regresión un Kernel dado por $\tilde{\kappa}(x_1, x_2) = \kappa(x_1, x_2) + \phi(x_1)\Sigma_\theta\phi(x_2)^T$, siendo Σ_θ la matriz de varianzas-covarianzas del estimado *a priori* del componente paramétrico θ .

[707: semipar2] Modelos semiparamétricos (Kernel+regresores): notas adicionales *****  09:00

Materiales: [SemiParametricRegression.pdf] [YouTube ▶]

Este vídeo amplía ciertos detalles del vídeo [706]. Los primeros dos minutos resumen rápidamente las ideas de dicho vídeo.

Luego, considera la estimación de la confianza de los parámetros explícitos estimados (matriz de varianzas-covarianzas a posteriori de $\hat{\theta}$), y considera los casos particulares de identificación “kernel” (proceso gaussiano) pura ($\Sigma_\theta = 0$) y los mínimos cuadrados recursivos ($\kappa(x_1, x_2) = 0$). Por último, enfatiza el caso particular de proceso gaussiano con media μ no nula que debe ser estimada (conocido como *ordinary kriging* en la literatura), que corresponde a $\phi(x) = 1$ de modo que $y(x) = z(x) + \mu + \epsilon$, siendo μ constante y ϵ ruido (blanco) de medida de una cierta varianza.

Caso de estudio

[708: semipcls] Modelos semiparamétricos (Kernel+regresores): clase Matlab para ejemplos ****  07:39

Materiales: [Cód.: KernelRegressionClass.m] [PDF] [YouTube ▶]

Este vídeo describe la estructura de una clase Matlab para ejecutar la regresión semiparamétrica (kernel de covarianza + componente paramétrico) cuya teoría se discute en los vídeos [706] y [707].

Esta clase será utilizada en el vídeo [709].

[709: semipcaso]

Modelos semiparamétricos (Kernel+regresores): caso de estudio

 14:40

Matlab

Materiales: [[CÓD.: semiparametricKernelRegressionTestsV2.mlx](#)] [[PDF](#)]

[YouTube ▶]

Este vídeo presenta un ejemplo de regresión semiparamétrica (kernel de covarianza + componente paramétrico) cuya teoría se discute en los vídeos [706] y [707]. Se utiliza la clase Matlab cuyo código se analiza en el vídeo [708]. Es aconsejable, por tanto, el visionar dichos vídeos previos para poder comprender el ejemplo aquí desarrollado.

Nota: sin descargar el fichero `.m` asociado al vídeo [708] en el mismo directorio que el `.mlx` aquí, el código `mlx` no funcionará correctamente.

En concreto, se genera una serie de datos y se comparan:

- Regresión lineal (recta de regresión, mínimos cuadrados)
- Regresión no paramétrica (kernel $\kappa(x_1, x_2)$ de media cero), a veces denominado *simple kriging*.
- Regresión semiparamétrica estimando media ($\kappa(x_1, x_2)$ y componente constante μ), denominado *ordinary kriging* en la literatura (bueno, cierto caso límite).
- Regresión semiparamétrica estimando un componente lineal $\theta_1 x + \theta_2$, adicional al componente de covarianza $\kappa(x_1, x_2)$.

Se comparan los estimados en media y varianza, así como los intervalos de confianza de los parámetros explícitos θ estimados.

Nota: la selección de hiperparámetros (parámetros de la función κ) ha sido más o menos “manual” para que salgan gráficas “bonitas”. Un enfoque más adecuado debería, posiblemente, incluir una optimización de hiperparámetros tal y como se considera en el vídeo [704].

Capítulo 21

Análisis multivariante de datos

El análisis multivariante trata de encontrar relaciones entre múltiples fuentes de datos, con tres objetivos (combinados):

1. *Clasificar*: por ejemplo, en una de las tres etiquetas (“normal”, “fallo 1” o “fallo 2”).
2. *Predecir (regresión)*: La mejor predicción de estas 3 variables $\{y_1, y_2, y_3\}$ a partir de estas 19 $\{x_1, x_2, \dots, x_{19}\}$ es $y = M_{3 \times 19}x$.
3. *Reducir dimensionalidad*:
 - Para representación o clasificación no supervisada: las muestras de 18 variables están muy correlacionadas, y pueden describirse, con error menor al 5%, a partir de la evolución de sólo 3 *componentes principales* o *variables latentes*.
 - Para regresión: sólo dos combinaciones de las 19 variables x parecen influir significativamente en las 3 variables y a predecir.

A partir de los años 2000, la gran disponibilidad de datos tanto en la red en general como en una instalación industrial en particular (cientos de sensores recogiendo miles de medidas por hora disponibles en archivos históricos) ha popularizado estas técnicas bajo el paradigma “*Big Data*”.

Este capítulo analiza dichas técnicas desde una perspectiva de “ingeniería industrial” o “ingeniería de sistemas” de forma introductoria, esto es, sin, quizás, el rigor y formalismo que sería el objetivo de un curso especializado de estadística sobre estas materias (que incorporaría tests sobre suficiencia del número de muestras, intervalos de confianza en las matrices de varianzas-covarianzas estimadas, tests de normalidad, etc.) y también evitando aspectos algorítmicos de eficiencia computacional que son de gran importancia si se desea realizar este tipo de análisis con, por ejemplo, *miles* de variables, merecedores de ser estudiados en profundidad por los lectores interesados en métodos numéricos o informática.

21.1 Análisis de componentes principales

El análisis de componentes principales (PCA, del término *principal component analysis* en lengua inglesa) es ampliamente usado en estadística, informática, clasificación, y también en identificación y monitorización de procesos para control (con variaciones para incorporar dinámica). Por ello se considera oportuno dedicar un tiempo a entender los conceptos básicos de esta metodología.

[710: pca0]

Análisis de Componentes Principales para ingenieros de control
(parte 1: motivación)

Materiales: [PCA.pdf]

**  14:59

[YouTube ▶]

Este vídeo discute las ideas que motivan el análisis por componentes principales:

- Planteamiento del problema: registro de 1000 muestras de 4 señales. Detectar las correlaciones entre los valores de las mismas (en el mismo instante).
- El análisis por componentes principales no distingue entre entradas y salidas. Las correlaciones elevadas entre variables se asociarán a “modelos” de los datos. Los componentes no correlados se asociarán a “causas de variabilidad” (perturbaciones o entradas exógenas).
- Utilidad para modelado, mejora de proceso, detección de fallos.
- Relaciones con otros campos científicos (informática, visión, economía, publicidad, control).

La ‘perpendicularidad’ entre componentes principales que se buscará está relacionada con que resulta conveniente para cálculos que las variables no tengan correlación, [960].

Nota: Se recomienda al lector revisar los conceptos básicos de álgebra de matrices (vídeos [matrix(09:38)], [matrixG(10:16)]), estadística (Sección B.2), y de valores singulares (Sección A.4, como mínimo los vídeos [svdOA(11:35)] y [svd2(17:10)]), para la comprensión correcta de los contenidos de los siguientes videos.

[711: pca1]

Análisis de Componentes Principales (parte 2: series de datos, teoría)

*** 18:59

Materiales:

[PCA.pdf]

[YouTube ▶]

Los “componentes principales” están relacionados con los valores y vectores propios de las matrices de varianzas-covarianzas (diagonalización). De hecho, la idea se utiliza para “tirar el dado” de distribuciones de probabilidad multivariable con correlación entre ellas dada: podría ser aconsejable visualizar de forma previa a este vídeo el vídeo [956] donde se introduce dicha idea. El análisis de componentes principales está motivado por el hecho de dar una interpretación similar a las matrices de varianzas-covarianzas obtenidas de datos experimentales.

En este vídeo se discuten:

- Formación de la matriz de datos
- Descomposición SVD de la matriz de datos. Relación con matriz de varianzas covarianzas.
- Definición de los componentes principales como cambio ortogonal de coordenadas. En esas nuevas coordenadas, los componentes principales no tienen correlación entre sí.
- La desviación típica de los componentes principales clasifica en “perturbaciones (causas de variabilidad)” las de varianza alta, y en “modelos” las combinaciones lineales asociadas a varianzas bajas.
- Se esboza la utilidad para clasificación
- Cuestiones relacionadas con escalados y normalización de los datos previa al cálculo de los componentes principales en aplicaciones prácticas.

21.1.1 Casos de estudio

[712: pca2ml]

Análisis de componentes principales (parte 3: ejemplo Matlab series de datos)

*** 10:37

Materiales:

[Cód.: pca2ml.zip] [PDF]

[YouTube ▶]

En este video se aborda, con Matlab, el análisis PCA de una serie de 1000 muestras de 4 variables.

Primero se normalizan a desviación típica unitaria y luego se forma la matriz de datos normalizados.

Se hace el SVD de dicha matriz, se calculan los componentes principales, su desviación típica y se dibujan las trayectorias temporales de dichos componentes. Obviamente, dada su importancia, existe un comando `pca` específico de la Toolbox de Estadística, pero realmente es SVD, ver vídeo [716].

Se esboza la posible utilidad de estos componentes:

- aproximación de la serie de cuatro datos en función de un número menor de “variables latentes” eliminando los componentes principales de baja desviación.
- posible uso para monitorización poniendo umbrales sobre los componentes en vez de sobre las salidas originales.

A simple vista parece que las 1000 muestras no son independientes entre sí en distintos instantes, las gráficas apuntan a la existencia de ciertos componentes frecuenciales que podrían dar lugar a interpretar estos datos como una serie temporal de tipo *ruido coloreado*. Este análisis dinámico, con estos datos, se aborda en los vídeos [756] y en el vídeo [766] se identifica un modelo en representación interna asociado a dicha dinámica.

Obviamente, el análisis de componentes principales es una metodología muy utilizada que tiene su propio comando en Matlab: `pca`, que realmente ejecuta internamente SVD.

[713: paises1] [Análisis de datos socio-económicos de una serie de paises \(1\): preprocesado](#)

Materiales: [[CÓD.:](#) paises123.zip] [[PDF](#)]

** 08:54

[YouTube ►]

Este es el primer vídeo de una serie de tres que abordan el caso de estudio de una serie de datos socio-económicos de 96 países, tomados de <http://halweb.uc3m.es/esp/Personal/personas/agrane/libro/100PEM.htm>, con permiso de la autora. Los siguientes son el [714] y [715] que abordan, propiamente, el análisis PCA; en este vídeo se detalla únicamente el preprocesado de los datos.

Las etapas del preprocesado son: lectura con el comando `textscan`, análisis de la asimetría de los histogramas (`skewness`) y transformación logarítmica para simetrizarlos; por último, se normalizan a media cero y desviación típica unidad, dado que los datos tienen orígenes y unidades muy diferentes.

El vídeo finaliza calculando la matriz de varianzas-covarianzas de los datos estandarizados (`cov`) que contiene las correlaciones dos a dos fuera de la diagonal. Los valores y vectores propios de dicha matriz darán lugar al cambio de variable que transforma a componentes principales ortogonales (interpretación geométrica), no correlados (interpretación estadística).

[714: paises2] [Análisis de datos socio-económicos de paises \(2\): cálculo de componentes principales \(svd, pca\)](#)

Materiales: [[CÓD.:](#) paises123.zip] [[PDF](#)]

*** 07:15

[YouTube ►]

En este vídeo se calcula el SVD de la matriz de varianzas-covarianzas Σ resultante de preprocesar (logarítmicamente) y estandarizar (media cero, vza. 1) unos datos de una serie de países. Dicho preprocesado se detalla en el vídeo [713], y se resume aquí en el primer minuto.

Los valores singulares (diagonal de S en la descomposición $\Sigma = U \cdot S \cdot V^T$, divididos por la raíz cuadrada del número de países menos 1 dan lugar a las desviaciones típicas de los

componentes principales, que son ΣV . Se analiza el porcentaje de variabilidad explicado por cada componente. Luego, se compara el resultado del `svd` con el comando `pca` siendo, obviamente, idéntico; esto ya se había visto en el vídeo [716].

La parte final del vídeo analiza el estadístico $T^2(x) = x^T \Sigma^{-1} x$ siendo x el dato estandarizado (media cero) de cada país. Se analiza su significado en términos de componentes principales, y se representa para la serie de datos de los países: aquéllos con un residuo T^2 alto son los que más lejos en probabilidad están de la media de los datos. Según la distancia sea en los primeros componentes principales o en los últimos podría tener una interpretación diferente en las aplicaciones; eso se discute en el vídeo [715], continuación de éste.

[715: pais3] Análisis de datos económicos de países (3): representación gráfica de los primeros y últimos componentes, interpretación

Materiales: [CÓD.: pais123.zip] [PDF]

*** 10:58

[YouTube ▶]

Este es el último vídeo que analiza un caso de estudio sobre componentes principales en unos datos económicos de 96 países (el primero es el [713] –preprocesado–, el segundo es el [714] –SVD, residuo T^2 –).

En este vídeo se visualizan e interpretan dos gráficas:

- Los dos componentes principales de *mayor variabilidad*. Es la gráfica 2D donde más se diferencian unos países de otros (de entre los posibles cambios de variable lineales y ortogonales). Se suele usar para clasificación no supervisada en aplicaciones y, en efecto, parece separar mucho a los países según un componente 1 asociado a un promedio de básicamente todos los datos (excepto ‘mujeres activas’ y ‘bosques’) – indicador económico– y un componente 2 ortogonal al primero, donde la tasa de natalidad, mujeres activas y superficie de bosque son los datos más influyentes en dicho índice.
- Los dos componentes de *menor* desviación típica. Podrían interpretarse como un modelo (una combinación lineal de los datos que es aproximadamente cero en la mayoría de muestras). Los países con mayor residuo en esas dimensiones serían aquéllos en los que la “relación entre los datos” es más diferente.

Esta separación entre los primeros componentes (debida a la *magnitud* inusual de las entradas –variables latentes en terminología estadística–) versus los últimos (debida a la diferente *relación* entre ellas, esto es, que siguen un *modelo* diferente) es lo que inspira la división de posibles fallos o errores en registros de datos (muestras anómalas) en “dentro de modelo” (primer caso) y “fuera de modelo” (segundo caso). El vídeo introductorio [776] detalla esta idea; el vídeo [777] la desarrolla teóricamente.

Nota: El vídeo [unfoldm(14:55)] también aborda en su primera mitad (hasta el instante 06:00) el análisis de componentes principales para descubrir que unos datos 4D realmente son un giro de un plano 2D donde está escrita la palabra “HELLO”. El PCA sólo realiza una “reorientación” de los datos, y no un “aplanamiento” por lo que está más limitado cuando se trata de usarlo para descubrir posibles estructuras **no lineales** subyacentes.

[716: pcaissvd] Comparacion PCA (statistics toolbox) con SVD ordinario: resultados idénticos

Materiales: [CÓD.: pcavssvd.mlx] [PDF]

** 03:40

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo, muy breve, comprueba que la descomposición en valores singulares (`svd`), que no necesita de ningún toolbox en particular instalado, produce el mismo resultado que la salida del comando `pca` de la Statistics and ML toolbox de Matlab. El vídeo [714] también verifica la coincidencia de la salida de estos dos comandos.

21.1.2 Componentes principales, versión Kernel (K-PCA)

[717: kpca] [Análisis de componentes principales: versión Kernel \(K-PCA\)](#)

***** 

Materiales: [KernelPCA.pdf]

Estas transparencias discuten la extensión del vector de características de unos datos con no-linealidades, expresando teóricamente la idea básica del vídeo [729]. Si el número de características (regresores) q es alto, entonces podría convenir traducir el problema PCA a su versión Kernel, y el detalle de cómo hacerlo es el objetivo de este material.

21.2 Regresión, correlaciones canónicas

21.2.1 PCR: regresión por componentes principales

[718: pcr] [Regresión por componentes principales \(PCR, Princ. Component Regression\)](#)

*****  17:26

Materiales: [PCR.pdf]

[YouTube ▶]

Este vídeo describe la metodología de regresión por componentes principales (principal component regression, PCR, en literatura en lengua inglesa). Básicamente, se trata de utilizar para un modelo de regresión (estimar y a partir de x , siendo ambos vectores) sólo los componentes de x con una varianza “suficientemente alta”. Estos componentes tienen unos parámetros estimados con una exactitud bastante razonable (la desviación típica de los parámetros estimados es inversamente proporcional a la varianza de la información –excitación–), y si desechamos los componentes con varianza baja entonces evitamos problemas de un estimado mal condicionado y sensible a ruido (que requerirían muchas más muestras). El PCR se puede usar como técnica de “regularización” (eliminar componentes de varianza muy pequeña, si se asume que son puro ruido), o bien como técnica de simplificación de modelos (escoger un número limitado de componentes de x con gran variabilidad y que expliquen suficientemente bien y).

Para la segunda de las tareas, no obstante, otras metodologías (PLS vídeo [721], CVA vídeo [724]) pueden estar más indicadas, dado que los componentes con más “varianza” de x no tienen por qué ser los que tengan más “covarianza” o “correlación” con la variable y a predecir.

Ejemplos Matlab de uso de PCR aquí explicado aparecen en los vídeos [720] (caso mono-variable 1 salida, estimar respuesta impulsional de un sistema discreto lineal) y [719] (caso multivariable, estimar 2 salidas a partir de 4 entradas).

[719: pcm] [Regresión por componentes principales \(PCR\): ejemplo Matlab](#)

*****  14:05

Materiales: [[CÓD.:](#) PruebaPCR3comp.mlx] [[PDF](#)]


[YouTube ▶]

Este vídeo presenta un ejemplo en código Matlab de regresión PCR (teoría en vídeo [718]), y de selección de un número de componentes que expliquen una fracción suficiente de la varianza de la variable a predecir, tanto sobre datos de entrenamiento como sobre datos de validación.

Otro ejemplo de regresión PCR aparece en el vídeo [685], y otro en un caso de identificación de la respuesta ante impulso de un sistema lineal discreto aparece en el vídeo [720].

[720: impulidPCR]

Identificación de la respuesta impulsional de un sistema discreto: regularización principal component regression

**** 

10:54

Materiales: [[CÓD.: impulseIDwithPCRregul.mlx](#)] [[PDF](#)]

[YouTube ▶]

Este vídeo utilizará regresión por componentes principales (principal component regression PCR, teoría en vídeo [718]) para regularizar un problema de estimación de la respuesta impulsional de un sistema discreto lineal.

De modo opcional (aquí se revisa la idea básica del PCR, que es una descomposición en valores singulares de la matriz de regresores por lo que este vídeo podría ser entendido relativamente bien) se recomienda visualizar el vídeo [718] arriba referido, o el ejemplo Matlab en el vídeo [719], para una explicación más detallada de toda la metodología.


El sistema y problema de identificación son los mismos que en los vídeos [736] y [737], que se resumen rápidamente en los dos primeros minutos aquí, cuya visualización previa se aconseja para entender mejor el análisis comparativo.

Nota: podría ser de interés visionar (opcionalmente, contenido avanzado) la regularización mediante un prior estocástico en este mismo problema discutida en los vídeo [745] y [747].

21.2.2 PLS: Mínimos cuadrados parciales

[721: pls1]

Mínimos cuadrados parciales [1]: Introducción y objetivos de la metodología PLS

***  08:58

Materiales: [PLS.pdf]

[YouTube ▶]

En este vídeo se presenta el planteamiento y motivación hacia el problema de mínimos cuadrados “parciales” (Partial Least Squares, PLS) donde se busca predecir un vector de variables y a partir de un conjunto de variables x .


La motivación básica arranca de que la regresión PCR (componentes principales de la entrada) no tiene por qué generar direcciones óptimas para la predicción de una salida (o vector de ellas), a no ser que lo que más varíe en la entrada sea lo que más correlación tiene con la salida. En un caso general, PLS trata de obtener direcciones que expliquen la salida en términos de propiedades (SVD) de la matriz de *covarianza* entre entradas y salidas. No obstante, las direcciones del SVD de la covarianza asociadas a la información (entrada al modelo) no son “independientes” (no correladas), por lo que se deben realizar operaciones adicionales para obtener componentes de entrada no correlados.

El objetivo del PLS es ordenar componentes de x en función de su “utilidad para predecir y ” y, viceversa, obtener qué componentes de “ y ” son mejor estimados a partir de x .

La exposición sobre PLS continua en los vídeos [722] y [723].

[722: plso]

Mínimos cuadrados parciales [2]: Preparación de datos, PLS con entrada ortonormalizada (pre-blanqueada)

****  12:48

Materiales: [PLS.pdf]

[YouTube ▶]


Este vídeo discute el detalle de lo que se denominan mínimos cuadrados parciales *ortonormalizados*, donde la información de entrada a un modelo ha sufrido un cambio de variable de modo que su matriz de varianzas-covarianzas sea la *identidad* (ortonormalización, pre-blanqueado: si $X = U_x S_x V_x^T$, siendo X la matriz de datos, entonces $X_{white} = \sqrt{N-1} V_x^T$ es la variable blanqueada buscada). En principio, ello no altera la capacidad de predecir y : si el estimador de mínimos cuadrados de y es Θx , con un cambio de variable $x = T x_{new}$,

el estimador será $\Theta' \cdot x_{new}$, siendo $\Theta' = \Theta T$, con el mismo resultado. Por tanto existe libertad total en elegir las unidades de X , en principio, y se escoge en PLS ortonormalizado una entrada preblanqueada.

Con estas entradas ortonormalizadas, el SVD de la matriz de covarianza $E(yx^T) = USV^T$ propone unas direcciones de X dadas por V que se asocian a las “covarianzas principales”, en la diagonal de S . La mejor predicción lineal de y será Sx : unos componentes mono-variables sin interacción (sin correlación cruzada) $y_i \approx \sigma_i x_i$, que cada uno contribuye a reducir la variabilidad total del error de predicción en σ_i^2 . La propuesta PLS para obtener modelos “simplificados” es despreciar aquellos componentes con σ_i pequeños.

Al obtenerse direcciones en entrada y salida a partir del SVD de la covarianza, las direcciones PLS no coinciden con las del PCA (componentes principales) ni de la entrada ni de la salida, pensando en unidades originales de las mismas.


Existen otras interpretaciones del PLS (SIMPLS,...), discutidas en el vídeo [723], que se aconseja ver a continuación de este, donde se hace una comparativa entre ellas y con PCR, para acabar de entender todas las ideas. Un ejemplo Matlab se aborda en el vídeo [728].

[723: plss] Mínimos cuadrados parciales [3]: ideas básicas SIMPLS, comparativa con O-PLS, PCR, discusión, conclusiones. *****  10:20
Materiales: [PLS.pdf] [YouTube ▶]

Este vídeo presenta las ideas básicas detrás de otros algoritmos PLS donde la matriz de entrada *no* es ortonormalizada previamente. Por ello, en la covarianza está “implícito” tanto la correlación de la salida y la entrada como las varianzas de las mismas. Por tanto, el PLS no ortogonalizado “pondera” no sólo la capacidad de x para predecir y , sino la propia varianza de x ... es un intermedio entre el PCR (no se considera la covarianza) y el PLS ortogonalizado del vídeo [722].

Matlab, en su comando `plsregress` implementa el algoritmo que se denomina SIMPLS. Empieza con las direcciones principales de la covarianza asociadas al valor singular máximo, y luego elimina determinados componentes de la matriz de covarianza. Los detalles no son objetivo del vídeo, y pueden ser consultados en la literatura. Existen otros algoritmos, como NIPALS, que persiguen objetivos similares.

21.2.3 CVA: análisis de variables canónicas

[724: cva1] Análisis de variables canónicas (canonical variate analysis, CVA): planteamiento y justificación ***  11:50
Materiales: [CVA1.pdf] [YouTube ▶]

Este vídeo aborda el planteamiento y justificación del significado de los objetivos del problema de regresión denominado “análisis de variables canónicas” (canonical variate analysis, CVA). Básicamente, la motivación es que el coeficiente de correlación monovariante es la covarianza dividido por el producto de desviaciones típicas. Con ello, si se escalan *entradas* y *salidas* a varianza unidad (desviación típica unidad), la covarianza coincide con la correlación y con el coeficiente del estimador de mínimos cuadrados de una variable dada la otra.

Este vídeo argumenta de que dichas ideas pueden ser generalizadas al caso multivariable. El CVA plantea un problema de regresión donde tanto la entrada como la salida están preblanqueadas (ortonormalizadas). Aunque el cambio de entradas es irrelevante (se mantiene el error de predicción inalterado), el cambio de variable en salidas *sí* cambia la ponderación relativa entre los distintos componentes de la salida (mínimos cuadrados ponderados). Si todo está a varianza unidad, el ajuste de todos los componentes de una salida

se ha reescalado para que el error de predicción tenga la misma importancia “relativa” (base 1) o “porcentual” en todos los componentes.

La covarianza de unas variables blanqueadas $\Sigma_{\tilde{y}\tilde{x}}$, en términos de las variables originales (x, y) es $\Sigma_y^{-1/2} \Sigma_{yx} \Sigma_x^{-1/2}$, que en el caso monovariable es el coeficiente de correlación y, en el caso multivariable, conforma lo que se denomina una *matriz de correlación*. El detalle algorítmico de estas ideas se continua en el vídeo [725].

[725: cva2] **Análisis de variables canónicas: diagonalización de la correlación, interpretación del resultado**

****  16:56

Materiales: [CVA2.pdf]

[YouTube ▶]

Este vídeo discute el detalle algorítmico del análisis en variables canónicas cuyas ideas preliminares se abordan en el vídeo [724]. Básicamente, en variables preblanqueadas, la matriz de varianzas covarianzas es:

$$\begin{bmatrix} I & \Sigma_{\tilde{y}\tilde{x}} \\ \Sigma_{\tilde{y}\tilde{x}}^T & I \end{bmatrix}$$

El análisis CVA propone realizar un segundo cambio de variable, además del preblanqueado, basándose en el SVD de $\Sigma_{\tilde{y}\tilde{x}} = USV^T$ (o sea, el mismo que el PLS ortogonalizado). Con ese segundo cambio, la matriz de varianzas-covarianzas conjuntas queda:

$$\begin{bmatrix} I & S \\ S^T & I \end{bmatrix}$$

y todos los elementos de la diagonal de S están entre cero y uno, teniendo unidades de coeficiente de correlación. Esos elementos son las correlaciones canónicas, y los cambios de variable determinan las “variables canónicas” asociando parejas de combinaciones de entradas y combinaciones de salidas conforme a su correlación.

La segunda mitad del vídeo discute una interpretación “gráfica” de los componentes canónicos. Básicamente, las variables canónicas pueden descomponerse, con una interpretación (informalmente asociada a conceptos de “control”, con un poco de imaginación) en:

1. Entradas, en dos grupos:
 - Grupo “no observable” que no tienen correlación con salidas, no sirven para predecirlas.
 - Grupo “observable” con correlación con las salidas.
2. Salidas, en dos grupos:
 - Grupo “no controlable” que no tienen correlación con entradas: las entradas no sirven para predecirlas y, viceversa, ellas tampoco sirven para predecir nada de la entrada.
 - Grupo “controlable” con correlación con las entradas y que, por tanto, si estas entradas fueran “manipulables” podrían ser modificadas (parcialmente, quizás) calculando los valores adecuados de la entrada (en variables canónicas, están relacionadas por una simple matriz diagonal).

[726: cvaplsrg] **Regularización de CVA y O-PLS**

***** 


Materiales: [gevpregulresum.pdf]

En estas transparencias se discute cómo incorporar regularización si los datos en CVA (tanto entradas como salidas) o en O-PLS (sólo entradas) están muy mal condicionados. Aparte de hacer un PCA previo y eliminar componentes con poca varianza, también puede incorporarse un elemento $c \cdot I$ a las matrices de varianza para mejorar su invertibilidad. El problema se presenta en un contexto donde se utiliza lo que se denomina valor propio generalizado, **eig(A,B)**, cuya definición, usos y propiedades en este y otros contextos se discute en el material [934].

21.2.4 Ejemplos de código

[727: plsm1]

Componentes principales en regresión (PLS y CVA): ejemplo matlab 1

****  15:15

Materiales:


[[Cód.:](#) pca2ml.zip] [[PDF](#)]

[[YouTube](#) ►]

Este vídeo presenta un ejemplo Matlab con la serie de datos del video [712], prediciendo 2 de las variables en función de las otras 2. Dado que un sólo componente principal permite explicar gran parte de la varianza de las 4 variables bajo análisis, resulta intuitivamente esperable que, si a partir de las variables “conocidas” puede inferirse (estimarse) el valor de ese componente principal, podrían predecirse con buena exactitud las dos variables “desconocidas”. Esa intuición es, en efecto, cierta. En el vídeo se analizan los resultados de PLS ortonormalizado y de CVA con las series de datos.

[728: plcvm]

Mínimos cuadrados parciales y variables canónicas: caso de estudio (Matlab)

****  21:36

Materiales:

[[Cód.:](#) PLSwine.zip] [[PDF](#)]

[[YouTube](#) ►]


Este vídeo analiza un conjunto de datos de 36 vinos del cual se dispone de cinco análisis “objetivos” (laboratorio) y de nueve calificaciones “subjetivas” (notas de cata). El objetivo es, claro, predecir las calificaciones “subjetivas” a partir de las “objetivas”, y analizar las “combinaciones de las medidas de laboratorio” que son capaces de predecir bien determinadas “combinaciones de las medidas de la nota de cata”. Se comparan PLS, PLS ortonormalizado y CVA con un número diverso de componentes para examinar su capacidad de predicción y el grado de correlación entre los componentes objetivos-subjetivos relacionados. No se compara con PCR dado que, técnicamente, resulta a priori menos indicado para regresión al no considerar covarianza o correlación entre entradas y salidas.

21.3 Extensión al caso no lineal

Nota: Hay muchos desarrollos en la literatura reciente sobre “reducción de dimensionalidad” e identificación en “variedades” (manifolds) con curvatura, en vez de los planos o hiperplanos (subespacios) que el PCA lineal puede únicamente manejar. Es una disciplina muy activa en el “machine learning” reciente. Sólo algunos ejemplos e ideas sencillas son presentados aquí.

[729: pcanlx]

Análisis de componentes principales: incorporación explícita de no-linealidades

***  10:09

Materiales:


[[Cód.:](#) nonlinpcacirculo.mlx] [[PDF](#)]

[[YouTube](#) ►]

En este vídeo se presenta la idea de extender el análisis de componentes principales al caso no lineal aumentando las muestras con funciones no lineales de las mismas. Tras una motivación/esbozo teórico de la idea, se presenta un ejemplo Matlab donde una serie de datos bidimensionales es generada con forma “circular”. El PCA lineal no consigue encontrar ninguna “estructura subyacente” en los datos, dado que dicha estructura no tiene forma de “recta que pasa por el origen” (subespacio de dimensión menor). Sin embargo, incorporando los cuadrados y producto de los elementos de cada dato se genera un vector de características (“feature vector”) $\phi(x)$ de dimensión cinco con el que sí que se averigua que determinado polinomio (combinación lineal de los monomios con los que se construyó ϕ) explica una estructura de los datos uni-dimensional... efectivamente, la ecuación de un círculo de radio r , esto es, $x^2 + y^2 - r^2 = 0$, es un polinomio de grado dos. La parte final del vídeo deshace las transformaciones para dibujar en las coordenadas originales el círculo identificado.

[730: unfoldm]

Desdoblamiento (unfolding) de curvas mediante LMI/SDP:
 ejemplo Matlab

*****  14:55

Materiales:

[\[Cód.: ManifUnf.zip \]](#)
[\[PDF \]](#)

[\[YouTube ▶\]](#)

En este vídeo se presenta un ejemplo Matlab de “desdoblamiento” o “estiramiento” de curvas (o variedades –manifolds– en un espacio de dimensionalidad elevada) para intentar descubrir estructuras no-lineales latentes en los datos. En el ejemplo, se dibuja la palabra “HELLO” en un plano, se dobla sobre un cilindro y se rota en un espacio cuatridimensional.

Primero, el PCA lineal es evaluado. Si el cilindro es de radio “elevado” (poca curvatura), entonces la proyección de mayor varianza (dos componentes principales de mayor valor propio asociado) encuentra dichas letras. Cuando la curvatura es mayor (radio más pequeño del cilindro) entonces el PCA ya indica que considerar la estructura como sólo bidimensional no parece adecuado (hay un tercer valor singular parecido al segundo que no deberíamos descartar); si, pese a todo, nos decantamos por reducir a 2D entonces se obtiene una versión muy deformada de las letras.

Como alternativa, se usa la metodología de desdoblamiento “isométrico” de variedades (manifolds) basada en desigualdades matriciales lineales (LMI), también denominadas en otra literatura como “semidefinite programming” (SDP). Primero se genera un *grafo* con k vecinos más cercanos y luego se plantea un problema de optimización de *maximizar* la suma de distancias entre los elementos del grafo que no estén directamente relacionados (equivalente “mecánico”: estructura articulada de barras con cargas eléctricas de modo que todo se “repele”). El resultado desdobra el cilindro y obtiene una copia casi exacta de las letras originales sin deformar.

Nota: la matriz de productos escalares introducida en el vídeo [unfoldm(14:55)] como consecuencia de la identidad $\|x - y\|^2 = x^T x + y^T y - 2x^T y$ es, de hecho, una matriz Kernel, que se discuten en la Sección 20.5. Las matrices Kernel tienen una interpretación “métrica” que este vídeo explota (aunque sin entrar en detalles, por brevedad).

Discusión final

[731: plsdisc]

PCA, PLS, CVA... conclusiones, críticas, limitaciones.

***  11:26

Materiales:

[\[PCADiscu.pdf\]](#)


[\[YouTube ▶\]](#)

Este vídeo realiza comentarios finales sobre la aplicabilidad, limitaciones y consecuencias del uso “masivo” de PCA, PLS, CVA con grandes series de datos, así como las causas de que pudieran producir resultados erróneos (mala selección de los datos, agrupamientos, mezclas de modelos, no-linealidad)...

21.4 Modelos para clasificación [introducción al problema]

[732: clasifintr1]

Ajuste de modelos para clasificación (1): planteamiento del
 problema (determinista)

*  15:42

Materiales:


[\[ModelosClasif1ESP.pdf\]](#)

[\[YouTube ▶\]](#)

* [LINK TO ENGLISH VERSION](#)

Este vídeo presenta el problema de ajustar θ en un modelo $f(x, \theta)$ para ajustar parejas de datos de entrenamiento (x_i, y_i) con $y_i \in \{0, 1\}$, esto es, clasificación binaria. El problema puede ser determinista (perro / no perro) o aleatorio (probabilidad de tener cierta mutación genética dado cierto marcador en un análisis de sangre). Este vídeo plantea el problema determinista, discutiendo las cuestiones que salen a relucir cuando es posible una clasificación “perfecta”, y las funciones de coste para clasificación “imperfecta” donde falsos positivos o falsos negativos sean inevitables.


La versión probabilística del enunciado de este tipo de problemas se aborda en el vídeo [733].

[733: clasifntr2]	Ajuste de modelos para clasificación (2): versión probabilística del problema	**  14:57
Materiales: [ModelosClasif1ESP.pdf]		[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo presenta la versión “probabilística” del problema de ajustar θ en un modelo $f(x, \theta)$ para ajustar parejas de datos de entrenamiento (x_i, y_i) con $y_i \in \{0, 1\}$, esto es, clasificación binaria. La versión “determinista” se discute en el vídeo [732], previo a éste.

El objetivo entonces es que el modelo estime la probabilidad de tener cierta salida a 1 dado $x...$ la versión más sencilla es maximizar la probabilidad de las muestras (maximum likelihood estimation) ajustando θ . Hay enfoques bayesianos más formales, fuera de la muy breve introducción al tema que se discute aquí.

[734: clasifNoLS]	Ajuste de modelos para clasificación (3): ¿valen mínimos cuadrados?	***  12:55
Materiales: [ModelosClasif1ESP.pdf]		[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute la aplicabilidad (bueno, más bien la NO aplicabilidad) de los mínimos cuadrados en clasificación; el vídeo es continuación de los vídeos [732] y [733].

Aunque los mínimos cuadrados para ajustar una función $f(x, \theta)$ a datos de entrenamiento $\{-1, +1\}$ podrían ser razonables como una primera opción para empezar con el problema, adolecen de varios inconvenientes:

- El coste es ‘simétrico’, pero algunos problemas se abordan con coste asimétrico,
- Lo único que realmente importa en el ajuste es el signo de $f(x, \theta)$ no su valor,
- Las distribuciones de probabilidad (o sus logaritmos) con resultados binarios (Bernouilly, o binomial si contamos varias repeticiones) no son proporcionales a ‘error al cuadrado’ como sí lo es el logaritmo de la distribución normal en mínimos cuadrados para ajuste de datos continuos (bajo ciertas suposiciones de error como ruido normal aditivo; ver vídeo [971]).

Por ello, una solución ‘naive’ de aplicar ‘regresión lineal’ o, en general, optimizar un índice mínimo-cuadrático para datos binarios, podría no tener las prestaciones deseadas o no tener una interpretación estadística formalmente adecuada.

Capítulo 22

Identificación de Sistemas Dinámicos

22.1 Modelos polinomiales discretos ARX/OE

[735: arxteo]

Identificación de modelos discretos lineales con estructura ARX por mínimos cuadrados

Materiales: [arxteo.pdf]



14:42

[YouTube ►]

En este vídeo se presentan los modelos en tiempo discreto Auto-Regresivos con entrada eXterna (ARX), cuya ecuación en diferencias se define como

$$y_{t+n} + a_{n-1}y_{t+n-1} + \cdots + a_0y_t = b_{m-1}u_{t+m-1} + \cdots + b_0u_t + \epsilon_t$$

modificada, también incluyendo la posibilidad de un retardo a la entrada, ver los materiales y vídeo para más detalle y notación. Los modelos ARX tienen una ecuación en transformada \mathcal{Z} dada por $y(z) = B(z)/A(z) \cdot z^{-\nu}u(z) + 1/A(z) \cdot \epsilon(z)$, donde ν es el retardo antes mencionado.

Se discute cómo plantear un modelo de predicción y la estructura para estimar los parámetros por mínimos cuadrados standard o recursivos.

También se comentan los comandos de la System Identification Toolbox, `arx` y `recursiveARX`, que realizan dicha identificación. El comando `arx` será discutido en detalle en el vídeo [738].

Nota: el caso particular $A(z) = 1$, esto es, la estimación de modelos sin denominador $G(z) = b_0 + b_1z^{-1} + b_2z^{-2} + \cdots + b_Nz^{-N}$ se discute, como ejemplo, en el vídeo [736].

22.1.1 Ejemplos Matlab

Caso de estudio: identificación respuesta impulsional discreta

[736: impulsid]

Identificación de la respuesta impulsional de un sistema LTI discreto: ejemplo Matlab

Materiales: [[Cód.:](#) impulseIDpart1.mlx] [[PDF](#)]



09:36

[YouTube ►]

Este video discute cómo identificar directamente la respuesta impulsional de un sistema discreto (los métodos que identifican respuestas ante impulso/escalón/frecuencia se denominan “no paramétricos” en alguna literatura; aquí gastaremos las fórmulas estándar de mínimos cuadrados, sin diferencia alguna con los métodos “paramétricos”). La teoría de todo esto aparece en el vídeo [679], cuya visualización previa se aconseja.

La primera parte del vídeo (hasta el minuto 04:40, aproximadamente) revisa la fórmula de *convolución discreta*, que dice que si la respuesta ante entrada $\{1, 0, 0, \dots\}$ es $\{h_0, h_1, h_2, \dots\}$ entonces la salida ante condiciones iniciales nulas verifica $y_k = \sum_{j=0}^k u_j h_{k-j}$. Con la convolución, se construye una matriz **toeplitz** que permite hacer regresión de mínimos cuadrados (pseudoinversa).

De hecho, se trataría de un modelo ARX como el visto en el vídeo [735], pero sin dinámica de “denominador” de la función de transferencia, esto es, un caso particular simplificado de esa clase de modelos (los modelos sin denominador $G(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}$ se denominan modelos FIR –finite impulse response–).

Se calcula la matriz de varianzas-covarianzas de los parámetros estimados (respuesta impulsional), y los intervalos de confianza y se representa gráficamente, según lo visto en el arriba referido vídeo [679].

Los valores estimados resultan muy sensibles a ruido, dado que hay pocos datos o/y no excitan suficientemente todas las direcciones necesarias; se necesitarían más datos o estimar menos parámetros (regularización).

Aunque en este vídeo se han formado todas las matrices y codificado las fórmulas para calcular intervalos de confianza, etc., en el minuto [08:20] se indica cómo se podría haber calculado los parámetros estimados con el comando **arx** de la System Identification toolbox.

[737: impulidreg]

Identificación de la respuesta impulsional de un sistema LTI discreto: regularización

10:37

[YouTube ▶]

Materiales:

[Cód.:
impulidwithregul.mlx
[PDF]

Este vídeo plantea cómo regularizar (disminuir sensibilidad al ruido) un problema de identificación de respuesta impulsional de un sistema lineal discreto invariante en el tiempo. En los dos primeros minutos se revisan rápidamente las ideas detalladas en el vídeo [736] donde se plantea el problema y se resuelve sin regularización; la visualización previa de ese vídeo podría ser aconsejable, sobre todo si no recuerdas bien la fórmula de convolución.

La primera opción de regularización revisada será el *truncar* la respuesta impulsional. Se hará la identificación sobre datos de entrenamiento con una longitud de respuesta desde 1 hasta 36, y se sugerirá aquél valor de longitud de respuesta (número de parámetros estimados) que minimice el error sobre unos datos de validación entrada-salida independientes de los de entrenamiento.

La segunda opción de regularización será el criterio de información de Akaike, que determina el factor de mejora del error que compensa introducir un nuevo parámetro adicional.

Ambos criterios recomiendan estimar alrededor de 9 parámetros, lo cual se hace y se propone como resultado final regularizado; tendremos más sesgo (bias) al usar un modelo menos expresivo (podremos acercarnos menos a la “verdad verdadera”), pero menos varianza (sensibilidad a ruido), con lo que la repetición del experimento da parámetros estimados similares (más similares que sin regularizar).

Como opción alternativa para regularizar, en el minuto [08:00] se plantea la regularización por “ridge regression”, minimizar $\|y - T\theta\|^2 + \lambda \cdot \|\theta\|^2$. La idea es que de ese modo los parámetros θ que no contribuyan a hacer pequeño el error sean empujados hacia cero. El valor de λ lo determinamos, como al principio, mediante la minimización del error sobre datos de validación.

En este ejemplo en particular, no funciona demasiado bien: los resultados son muy parecidos al estimado no regularizado, porque en cuanto empezamos a “empujar” hacia cero sube muy rápidamente el error.


Nota: Este mismo problema se utiliza en otros materiales para evaluar las prestaciones de otras dos metodologías de regularización:

- El vídeo [720] utiliza los valores singulares (y las direcciones asociadas) de la matriz de regresores T para hacer regularización por “principal component regression”, cuya visualización se aconseja (quizás, previamente, mirad la teoría en el vídeo [718]).
- Otra opción de regularización basada en procesos estocásticos gaussianos se aborda en los vídeos [745], [746] y [747]; los contenidos son de mayor dificultad, no es necesario visualizarlos para entender correctamente las ideas aquí planteadas.

Otros ejemplos

[738: arxml] Identificación de modelos discretos ARX por mínimos cuadrados no recursivos: ejemplo MATLAB

Materiales: [[CÓD.:](#) recursivetestdynamicarxvideo.mlx] [[PDF](#)]

***  17:07

[YouTube ▶]

En este vídeo se genera una serie de datos con ruido de proceso tal y como postula la definición de un modelo ARX, cuya teoría se discute en el vídeo [735].


A continuación, se generan las matrices para hacer regresión por mínimos cuadrados no recursivos y se estiman, mediante la pseudoinversa de la matriz de regresores, los parámetros del modelo, comprobando su exactitud. Se calcula, asimismo, la matriz de varianzas-covarianzas de los parámetros estimados.

Después, se realizan los mismos cálculos con el comando `arx` de la System Id. toolbox, comprobando la coincidencia de resultados.

La última de las ideas exploradas en el vídeo es analizar qué ocurre si los datos no han sido generados “exactamente” con un ruido blanco filtrado por el denominador de la función de transferencia a identificar (ver notas y vídeo [735] arriba). En ese caso, el estimado ARX es *sesgado*. Dado que, de un modo u otro, esa situación *siempre* ocurrirá en la práctica, para validar/invalidar los resultados, se propone analizar el ajuste con datos de validación y examinar la correlación del residuo consigo mismo en diferentes instantes (autocorrelación) y con la entrada. Para ello, se puede utilizar el comando `resid` de la System Id. toolbox.

[739: arxrem] Identificación de modelos lineales ARX: mínimos cuadrados recursivos (Matlab)

Materiales: [[CÓD.:](#) recursivetestdynamicarxvideo.mlx] [[PDF](#)]

***  13:12


[YouTube ▶]

En este vídeo se continúa el ejemplo del vídeo [738], esta vez obteniendo los mismos resultados pero utilizando la implementación de mínimos cuadrados recursivos para los modelos ARX, que Matlab puede realizar con el comando `recursiveARX` de la System Id. toolbox.

En el vídeo, se inicializan los vectores de parámetros en media y en varianza, y se ejecuta el algoritmo recursivo. Con factor de olvido 1, los resultados cuando se han incorporado todos los datos son iguales a los de la versión no recursiva. Con factor de olvido menor a 1, la desviación típica de los parámetros estimados es mayor (se olvida parte de la información, la más antigua), y además dichos estimados son mucho más sensibles a ruido. Por último, se comprueba que cuando el ruido no es generado por un modelo ARX (filtrado por denominador de FdT), el estimador recursivo también, evidentemente, resulta sesgado (del mismo modo que lo era el no recursivo).

[740: oeml] Identificación de modelos lineales discretos Output-Error (OE): ejemplo Matlab

Materiales: [[CÓD.:](#) recursivetestdynamicOE.mlx] [[PDF](#)]

*** / ****  18:58

[YouTube ▶]

En este vídeo se discute mediante un ejemplo Matlab la identificación de modelos lineales en tiempo discreto con estructura output-error (OE), esto es aquéllos cuya dinámica es definida por:

$$y(z) = \frac{B(z)}{F(z)} z^{-\nu} u(z) + \epsilon(z)$$

siendo ϵ una secuencia de ruido blanco que podríamos denominar “ruido de medida”. El código es muy similar al de los vídeos [738] y [739], por lo que se recomienda la visualización previa de los mismos.

En este vídeo, primero, se evalúa el funcionamiento de la versión no recursiva del algoritmo, implementado en el comando `oe`. Los resultados son satisfactorios con datos generados tal y como estipula la definición. Resulta importante destacar que, incluso cuando el ruido ϵ es “coloreado” se logra identificar un modelo razonablemente exacto de la función de transferencia discreta, aunque el ajuste y la autocorrelación de los residuos dan a indicar que hay dinámica en dicho residuo que podría ser modelada con otra estructura de modelo (`arx` o `bj`). Esto es de interés en la práctica, dado que OE intenta que la simulación en “bucle abierto” de la respuesta ante la entrada conocida u sea lo más parecida a los datos, y lo consigue aunque el ruido sea coloreado (eso sí, se necesitan muchas muestras).

En la segunda parte del vídeo (a partir de [08:31]), se analizan los resultados conseguidos por la implementación recursiva `recursiveOE`. Debido a la no-linealidad del problema de identificación, los distintos gradientes, Hessianos, etc. necesarios para llevar a cabo la identificación hacen que la versión recursiva necesite muchísimas más muestras para estimar los parámetros correctos. Si se incorpora un factor de olvido de 0.999, el resultado converge más rápido a los parámetros correctos cuando el ruido de medida es blanco tal y como estipula la definición OE. Sin embargo, cuando el ruido de medida es coloreado, el algoritmo recursivo converge a parámetros incorrectos y es muy sensible a ruido con olvido menor a 1. Por ello, ante ruido coloreado (ruido de proceso) ϵ significativo, no hay demasiadas garantías de un funcionamiento correcto en aplicaciones de `recursiveOE`, pese a que la versión no recursiva sí resultaba razonablemente prometedora. ****

[741: t1stf] Mínimos cuadrados TLS para identificar función de transferencia (SISO) **** 21:28 [PLAY ▶]

Este vídeo ilustra el TLS del vídeo [692] en un contexto de sistemas dinámicos, buscando relaciones entre variables de entrada y salida en diferentes instantes. Utiliza mínimos cuadrados totales (restando ruido de medida de y en la matriz de varianzas-covarianzas) para estimar un modelo en función de transferencia (monovariable) con menos “sesgo” que un ARX. De todos modos, la optimización de `oe` obtiene un mejor resultado. Otros refinamientos son necesarios, dando lugar a los métodos de subespacio (Capítulo 23).

Nota: Matlab tiene una interfaz gráfica para la identificación de sistemas dinámicos (`SystemIdentification`). Por ejemplo, un breve tutorial del mismo realizado por mi compañero J.M. Herrero aparece en <https://media.upv.es/#/portal/video/ec89d9d7-e22f-4b8d-a998-7ed86de75430>.

22.1.2 Identificación de series temporales sin entrada

[742: ibexabs] Identificación serie Ibex 35: modelo autoregresivo arx, ejemplo **** 10:35
Matlab
Materiales: [Cód.: ibexcasestudy.zip] [PDF] [YouTube ▶]

En este vídeo se plantea, a partir de unos datos históricos de 10 años del IBEX35, intentar estimar un modelo lineal auto-regresivo: $y_k = \theta_1 y_{k-1} + \theta_2 y_{k-2} + \dots + \theta_5 y_{k-5}$. El vídeo

[744] también presenta otro caso de identificación de serie temporal que podría ser de interés visionar tras haber analizado el caso aquí desarrollado.

Primero se hace un `detrend` para eliminar la tendencia a “largo plazo” creciente; el objetivo es ver si, aparte de eso, se puede identificar algún modelo que permita decisiones de inversión beneficiosas a “corto plazo”. El modelo se estima con los comandos `iddata` y `arx`. El comando `compare` presenta un ajuste a datos (de validación) muy bueno (90%) que parece indicar que hemos obtenido un modelo fiable... no obstante, un análisis más detallado de errores e intervalos de confianza de parámetros estimados hace que realmente no haya ninguna mejor significativa respecto al modelo “trivial” $y_k = y_{k-1}$ (el IBEX mañana estará más o menos igual que hoy). El vídeo [743] intenta mejorar (infructuosamente) el modelo usando la serie temporal de “incrementos”.

[743: ibexinc]

Identificación serie Ibex 35: modelo autoregresivo incremental (paso-alto)

*** 06:08

Materiales: [[Cód.:](#) ibexcasestudy.zip] [[PDF](#)]
[YouTube ►]

Este vídeo es continuación del vídeo [742]. Dado que los beneficios de una inversión son sobre los “incrementos” de cotización, se hace un `diff` y se intenta obtener un modelo que explique esas variables incrementales. En control, sería equivalente a usar un filtro paso-alto sobre los datos para enfatizar el ajuste a altas frecuencias (“corto plazo” para los economistas).

El resultado es, como lo fue en el vídeo [742], infructuoso. El análisis del modelo obtenido confirma que no hay diferencias significativas respecto a las predicciones que hace el modelo “trivial”, $\Delta y_k = 0$, esto es, que mañana estaremos igual que hoy más o menos (paseo aleatorio lineal); no nos vamos a hacer millonarios con este código.

[744: idst]

Simulación e Identificación Serie Temporal, modelo ARMA

**** 13:36

Materiales: [[Cód.:](#) idserietemporalx2.mlx] [[PDF](#)]
[YouTube ►]

En este vídeo se presenta cómo simular y, luego, cómo identificar una serie temporal (sin entrada u) responsable de un ruido de proceso observado en un hipotético experimento donde no haya incrementos de variables manipuladas u conocidos.

La simulación de procesos estocásticos continuos se realiza generando una secuencia discreta de entrada de ruido (de varianza 1) que debe ser dividida por la raíz cuadrada del período de muestreo.

La identificación a partir de datos usa el comando `armax` de la System ID toolbox, que al no tener entrada identifica un modelo $A(z)y(z) = C(z)e(z)$ donde $e(z)$ es un ruido blanco que excita a $\frac{C(z)}{A(z)}$ para generar el ruido coloreado observado. Dado que los datos habían sido generados por un modelo de primer orden, el resultado de utilizar el comando con un modelo de segundo orden tiene un buen ajuste, pero mucha desviación típica en parámetros estimados... el ajuste de primer orden tiene el mismo ajuste pero unos parámetros estimados mucho más precisos.

Para comprobar que no dejamos dinámica no modelado, se usa el comando `resid` para dibujar el gráfico de autocorrelación de los residuos.

El último de los contenidos del vídeo aborda la obtención de un modelo continuo, usando `d2c` y, para normalizar a una excitación de ruido blanco continuo de densidad espectral de potencia 1, se corrige multiplicando la función de transferencia continua por la raíz cuadrada del período de muestreo y la desviación típica de $e(z)$ identificada.

22.1.3 Regularización con métodos Kernel (ident. rep. impulso)

[745: impulkerT]
09:41

Identificación de respuesta impulsional discreta con regularización Kernel (proc. gaussiano): Teoría

***** 

Materiales: [[CÓD.:](#) WienerAndIntegratedWienerForImpulseID.pdf]

[YouTube ▶]

Este vídeo aborda la regularización de la identificación de la respuesta ante impulso de un sistema lineal invariante en el tiempo (discreto) con un prior gaussiano descrito por un cierto Kernel de covarianza.

La respuesta ante impulso es la salida $\{h(t)\}$, $t = 0, 1, 2, \dots$ ante la secuencia de entrada $\{1, 0, 0, 0, \dots\}$.

Un enfoque “clásico” de mínimos cuadrados a este problema de identificación se detalla en el vídeo [736]. La regularización por truncación se aborda en el vídeo [737] y la regularización por componentes principales se aborda en el vídeo [720]. Su visualización previa es de interés para comprender con una mejor perspectiva el caso de estudio que aquí se presenta.

Dado que el enfoque está basado en los procesos estocásticos gaussianos, se recomienda visualizar previamente los conceptos básicos en, como mínimo, los vídeos [201], [202], [208] y [213] si no se está familiarizado con las ideas presentadas en la parte final del vídeo.

La primera parte del vídeo revisa las bases teóricas preliminares: planteamiento del problema, fórmula de convolución discreta, y mínimos cuadrados hasta minuto [01:40], desde ese instante hasta [02:40] se revisan las fórmulas de mínimos cuadrados recursivos (detalle en vídeo [686]).

La idea principal es el hecho de que la respuesta impulsional $h(t)$ de un sistema (muestreado a una frecuencia suficientemente alta, posiblemente, si los datos vienen del mundo continuo) no es algo “aleatorio”, sino una función relativamente “suave” con poca probabilidad de que tenga grandes cambios a “alta frecuencia” (que, sin embargo, aparecen por culpa del ruido en problemas sin regularizar, vídeo [736]).

Las funciones aleatorias del tiempo donde se especifica una medida de probabilidad de que haya variabilidad diferente según la “frecuencia” pueden ser modeladas como procesos estocásticos gaussianos con una cierta función de covarianza $\kappa(t_1, t_2)$.

Como el tiempo es discreto, hablaremos de $\kappa(i, j)$, $i = 0, \dots, n$, $j = 0, \dots, n$ siendo $n = \text{orden} - 1$ con orden el número de parámetros a estimar.

En el caso estacionario $\kappa = \bar{\kappa}(|t_2 - t_1|)$ el reparto de la varianza en la frecuencia es la densidad espectral de potencia.

Por tanto, esta idea acabará inspirando un “prior” de $h(t)$, de la forma $\mathcal{N}(0, \Sigma)$ para los mínimos cuadrados recursivos donde Σ está generado por la función κ , siendo esa estructura de Σ denominada matriz “kernel” de covarianza.

La suavidad de $h(t)$ puede ser generada por un $\bar{\kappa}$ que decrezca conforme la separación entre instantes aumente, la estabilidad con una cierta tasa (si se supone conocido a priori) deberá modelarse con un kernel no estacionario, donde κ dependa no sólo de la diferencia de tiempos sino de los tiempos en sí. La estabilidad se codificará en el prior mediante una cierta transformación exponencial a los argumentos de κ .

Ejemplos concretos de κ se abordan en el vídeo [746], continuación de éste; un ejemplo de identificación propiamente dicha se aborda en el vídeo [747].

[746: impulkerM1]
12:52

Identificación de respuesta impulsional discreta con regularización Kernel: ejemplo modelos a priori

***** 

Materiales: [[CÓD.:](#) WienerAndIntegratedWienerForImpulseID.mlx] [[PDF](#)]

[YouTube ▶]

Este vídeo aborda la regularización de la identificación de la respuesta ante impulso de un sistema lineal invariante en el tiempo (discreto) con un prior gaussiano descrito por un cierto Kernel de covarianza.

Las ideas teóricas principales se discuten en el vídeo [745], que se resume en este vídeo hasta el instante [01:40]. Su visualización (asi como otros vídeos con material preliminar ahí referidos) es necesaria para poder comprender bien este vídeo para una audiencia no familiarizada.

La idea principal es el hecho de que la respuesta impulsional $h(t)$ de un sistema (muestreado a una frecuencia suficientemente alta, posiblemente, si los datos vienen del mundo continuo) no es algo “aleatorio”, sino una función relativamente “suave” con poca probabilidad de que tenga grandes cambios a “alta frecuencia”.

Las funciones aleatorias del tiempo donde se especifica una medida de probabilidad de que haya variabilidad diferente según la “frecuencia” pueden ser modeladas como procesos estocásticos gaussianos con una cierta función de covarianza $\kappa(t_1, t_2)$. Como el tiempo es discreto, hablaremos de $\kappa(i, j)$, $i = 0, \dots, n$, $j = 0, \dots, n$ siendo $n = \text{orden} - 1$ con orden el número de parámetros a estimar.

Este vídeo plantea ejemplos concretos de $\kappa(i, j)$:


- $\kappa = M \cdot e^{(j-i)^2/2/\tau^2}$, squared exponential o “filtro Gaussiano”
- El proceso de Wiener (paseo aleatorio) con transformación exponencial de argumentos $\kappa = M \cdot \min(e^{-i/\tau}, e^{-j/\tau})$.
- El proceso de Wiener integrado (aceleración ruido blanco), también con transformación exponencial.

Los procesos estocásticos Wiener y Wiener integrado son también explorados en el video [242] en un contexto de filtrado/ridge regression, cuya visualización podría ser de interés. Nótese, sin embargo, que el planteamiento del problema es muy diferente: en ese vídeo se busca interpolar/extrapolar muestras faltantes, y ese no es el caso aquí.

En concreto, se ponen ejemplos de hiperparámetros M y τ y se analiza la desviación típica a priori (raíz cuadrada de diagonal de $K = \kappa(0 : n, 0 : n)$), los componentes principales Karhunen-Loève (vídeo [213]) y la forma que tienen unas cuantas realizaciones de los mismos (vídeo [208]), para entender su validez como estimados “a priori” de respuestas impulsionales de sistemas estables.

La conclusión es que estamos haciendo regularización porque unos “pocos” componentes principales explican gran parte de la variabilidad a priori de $h(t)$, y las realizaciones parecen “intuitivamente razonables”.

El ejemplo de identificación propiamente dicha con estos procesos estocásticos como prior (mediante los kernels asociados aquí explorados), se aborda en el vídeo [747], continuación de éste.

<p>[747: impulkerM2] 10:58</p>	<p>Identificación de respuesta impulsional discreta con regularización Kernel: ejemplo Matlab</p> <p>Materiales: [CÓD.: kernelimpulseIDv2b.mlx] [PDF]</p>	<p>***** </p> <p>[YouTube ▶]</p>
------------------------------------	--	---

Este vídeo aborda la regularización de la identificación de la respuesta ante impulso de un sistema lineal invariante en el tiempo (discreto) con un prior gaussiano descrito por un cierto Kernel de covarianza. Es continuación de los vídeos [745] y [746], que deberían ser visualizados previamente, aunque aquí se dedica hasta el instante [02:10] a revisarlo.

La idea principal es el hecho de que la respuesta impulsional $h(t)$ de un sistema (muestreado a una frecuencia suficientemente alta, posiblemente, si los datos vienen del mundo continuo) no es algo “aleatorio”, sino una función relativamente “suave” con poca probabilidad de que tenga grandes cambios a “alta frecuencia”, que puede ser modeladas como proceso estocástico con una cierta función de covarianza $\kappa(t_1, t_2)$.

Un enfoque “clásico” de mínimos cuadrados a este problema de identificación se detalla en el vídeo [736]. La regularización por truncación se aborda en detalle en el vídeo [737]; esta regularización será comparada con la opción Kernel discutida aquí. Las ideas principales de estos vídeos se resumen desde el instante [02:10] al [03:05].

Se compararán los resultados con Kernels exponencial cuadrático, Wiener y Wiener integrado (con transformación $e^{-t/\tau}$), descritos en el vídeo [746].

En todos ellos hay unos *hiper*parámetros (es la nomenclatura usual en literatura, porque llamaremos “parámetros” a los elementos de la respuesta impulso a estimar) de desviación típica y constante de tiempo τ . Se escogieran los parámetros que menor error ante datos de validación produzcan, siendo buscados con optimización “fuerza bruta” (por simplicidad), siguiendo las ideas del vídeo [704], cuya visualización previa es aconsejable.

Se comprueba que el kernel exponencial cuadrático filtra los datos no regularizados pero no fuerza que la respuesta impulso identificada tienda a cero, porque era estacionario (varianza constante). Los no estacionarios sí que producen estimados regularizados que tienden a cero; el Wiener integrado (doble integrador de ruido) produce la respuesta impulsional más “suave” de todos, como era de esperar intuitivamente. El ajuste de este último sobre datos de validación es bueno, mejor en muchos casos que la truncación.

22.2 Identificación de sistemas en tiempo continuo

[748: procest] Identificación con ‘procest’ de modelos de primer y segundo orden + retardo

Materiales: [[CÓD.:](#) Identificar1ordenyretrosoPROCEST.zip] [[PDF](#)]

*** 07:28

[YouTube ▶]

En este vídeo se usa el comando **procest** de la System Identification toolbox para estimar un modelo de primer orden + retardo (introducción al retardo en vídeo [119]) ante los mismos datos experimentales usados en el vídeo [154] para ajustar “manualmente” dicho modelo. Este tipo de modelos son populares en procesos industriales, y existen metodologías de diseño de PID adaptadas a ellos (por ejemplo, ver vídeo [403]).

El comando **present** presenta el modelo obtenido, con los valores nominales y la desviación típica de los parámetros estimados. El vídeo argumenta que la interpretación de la desviación típica debe hacerse con mucha precaución, dado que puede llevar a conclusiones incorrectas si las suposiciones de la teoría no se cumplían en el experimento que generó los datos.

La parte final del vídeo identifica, también con **procest** un modelo de segundo orden + retardo. La desviación típica elevada de la constante de tiempo no dominante (casi la tercera parte del valor nominal estimado) alerta de que la mayor complejidad del modelo podría ser innecesaria: en efecto, con los mismos datos apenas mejora el ajuste resultante del comando **compare**. En el vídeo también se discute sobre si ejecutar **compare** con condiciones iniciales nulas o bien dejarle que ajuste las condiciones iniciales por si acaso los datos inicialmente no estaban en equilibrio (que es la opción activa por defecto).

Otro ejemplo de uso de **procest** para identificar modelos de primer orden + retardo aparece en el vídeo [156], y para identificar modelos de perturbaciones (para prealimentación) aparece en el vídeo [479].

22.3 Identificación directa de parámetros físicos

[749: identga] Identificación experimental con algoritmo genético de parámetros de modelo masa-muelle-amortiguador

Materiales: [[CÓD.:](#) identificaGA.zip] [[PDF](#)]

*** 19:49

[YouTube ▶]

Este vídeo carga un fichero de datos (formato .mat) supuestamente experimentales, obtenidos en laboratorio, de un sistema mecánico del que se quiere estimar los parámetros físicos de un modelo masa-muelle-amortiguador lineal.

El vídeo describe una primera etapa de preprocesado de datos, donde se elimina el segmento inicial con efecto de condiciones iniciales y grandes desplazamientos (que podrían excitar no-linealidades presentes en casi todos los sistemas físicos), y se pone a “cero” alrededor del punto de funcionamiento donde se desea identificar el sistema lineal, como se hace de forma “teórica” en los ejercicios de linealización.

Luego, presenta el modelo de simulación paramétrico y la función que con `ode45` hace la integración numérica para unos parámetros datos (interpolando la entrada linealmente en puntos intermedios si fuera necesario para `ode45`).

Las salidas simuladas y experimentales se comparan, estableciendo un índice de coste de mínimos cuadrados que debe ser lo más pequeño posible.

La optimización en sí se ejecuta con un algoritmo genético `ga` de la Global Optimization toolbox de Matlab. Es un método global que, a cambio de ser mucho más lento en tiempo de ejecución (importante en problemas con muchas variables a optimizar), no se queda atascado en mínimos locales como otras rutinas (`fminunc` o `fmincon`) basadas en búsqueda local de dirección de máximo descenso; estas ideas se discuten con más detalle en el vídeo [927], aconsejable si no estás familiarizado con estas técnicas.

Se superpone el modelo identificado a datos de ajuste y de validación y se considera suficientemente preciso, con lo que se da por preliminarmente válido el modelo obtenido.

La extensión a identificación experimental de un modelo no lineal para estos mismos datos se aborda en el vídeo [750], continuación de éste. También se discute en el referido vídeo la posible utilización de métodos de búsqueda local (`fmincon`) para abordar este tipo de problemas.

Nota: en problemas complicados, existe una opción para acelerar los algoritmos genéticos usando múltiples núcleos de la CPU, ver documentación sobre la opción ‘UseParallel’.

Identificación experimental de parámetros físicos usando la System Identification Toolbox de un problema similar se aborda en el vídeo [751]. las ideas de este vídeo están detrás de muchos problemas de ingeniería; en particular, la optimización de parámetros de controladores, como se discute en los vídeos [585], [592], [593], etc.

[750: identganl]

Identificación experimental modelo no-lineal

masa-muelle-amortiguador: algoritmo genético, `fmincon`

Materiales: [[CÓD.:](#) identificaGANoLineal.mlx] [[PDF](#)]

**** 17:33

[YouTube ▶]

Este vídeo realiza, con sencillas modificaciones, la adaptación del vídeo [749] para identificar un modelo dinámico masa-muelle-amortiguador con un muelle no lineal $F_{muelle} = k_1 \cdot x + k_2 \cdot x^2$, siendo x el incremento de longitud respecto a la que el muelle tenga en el punto de operación alrededor del cual se identifican parámetros. También se prueba una no-linealidad cúbica, por comparar.

Como novedad metodológica, no obstante, se propone el uso de búsqueda “local” con el comando `fmincon` dándole como “semilla” (estimado inicial) un valor arbitrario o, mejor, el valor propuesto por el algoritmo genético. El comando `fmincon` realiza optimización basándose en gradientes, o en descenso local numérico (Nelder-Mead), o... bueno, detalles no son importantes aquí. De todos modos, estos algoritmos locales (más rápidos que genéticos, pero más propensos a quedarse atascados) no parecen progresar más que el genético en este problema concreto. Para aclarar diferencias entre búsqueda local (relativamente rápida) y búsqueda global (lenta), ver vídeo [927].

Realmente todo es muy similar al referido vídeo del caso lineal, pero se observa que, como era de esperar, un modelo no lineal parece ajustar mejor en entrenamiento y en validación.

Nota: también se puede usar ejecución en paralelo en máquinas multicore (casi todas hoy en día) si se tiene el Parallel Computing Toolbox instalado, con la opción “UseParallel”; eso no se comenta en el vídeo, pero sí en el MLX/PDF de materiales; en problemas sencillos a veces el sobrecoste de cargar múltiples copias de Matlab en memoria hace que no compense, al menos la primera vez que se ejecuta.

Rutinas globales/locales similares a las aquí propuestas están en el núcleo de las funciones de la System Identification Toolbox para tareas de identificación “grey-box” caja-gris de parámetros físicos como `idnlgrey`, como por ejemplo las usadas en el vídeo [751]. También, claro, están detrás de muchos problemas de ingeniería; en particular, la optimización de parámetros de controladores, como se discute en los vídeos [585] (usando add-ons de optimización de Simulink), o [592], [593], usando `ode45` directamente, como aquí.

[751: idnlg] Identificación sistema masa-muelle no-lineal: Matlab, idnlgrey

*** 18:04

Materiales: [[CÓD.:](#) idnlg.zip] [[PDF](#)]

[YouTube ▶]

En este vídeo se discute cómo utilizar las rutinas de la System Identification Toolbox de Matlab para identificar los parámetros de un modelo en representación interna no lineal de un sistema masa-muelle-amortiguador con cuatro parámetros físicos ($\theta_1, \dots, \theta_4$):

$$\begin{aligned}\frac{dp}{dt} &= v \\ \frac{dv}{dt} &= -\theta_1 p - \theta_2 p^3 - \theta_3 v + \theta_4 u + \xi \\ y &= \begin{pmatrix} p \\ v \end{pmatrix} + \psi\end{aligned}$$

a partir de lecturas de dos salidas (posición, velocidad) y una entrada. Este tipo de modelos es denominado “nonlinear grey-box” (*nlgrey*).

Sin utilizar la toolbox de identificación, pero sí la de algoritmos genéticos (global optimization toolbox), el vídeo [749] aborda un problema bastante similar, donde sólo se identifica un modelo lineal masa-muelle-amortiguador; depende de tu familiaridad con estas ideas, podría ser conveniente que lo visualizaras previamente.

En la primera parte del vídeo, se generan los datos con `ode45` y una ecuación de estado del proceso “real”. Para más detalles sobre el uso y opciones de `ode45`, ver vídeo [32] introductorio a la simulación.

Obviamente, ese proceso “real” no es conocido, ni el ruido de proceso $\xi(t)$ que se añade a la ecuación de estado para acercar más a una situación realista la simulación. También se añade ruido de medida aleatorio ψ a las muestras de las salidas.

En la segunda parte del vídeo (a partir del instante [06:04]), se usan las muestras de entrada, posición y velocidad para ajustar un modelo con la estructura de la ecuación anterior (que debe ser creada en un fichero .m separado), mediante el comando `idnlgrey` que inicializa la estructura, parámetros iniciales y estados iniciales, y la aplicación posterior del comando `nlgreyest` que, alimentado con un `iddata` adecuado, realiza propiamente el ajuste de los parámetros, para intentar que la simulación en bucle abierto ante la entrada proporcionada coincida con los datos medidos (minimizando error cuadrático).

Es importante remarcar que `nlgreyest` hace un ajuste con optimización no-lineal iterativa que, con modelos más complejos o no linealidades más severas, podría tardar mucho en converger o incluso no obtener una solución aceptable (el optimizador puede quedarse atrapado en un mínimo local espúreo inaceptable). Si deseamos estimar un modelo en representación interna *black box* de un sistema lineal, podemos optar por comandos de identificación subespacio `n4sid` (tiempo discreto) o `ssest` (continuo). Ver, por ejemplo, el vídeo [766].

Nota: La identificación de modelos en espacio de estados requiere, simultáneamente, estimar (observar) la trayectoria de las variables internas no directamente medibles. Incluso los parámetros podrían ser variantes en el tiempo. Se recomienda la visualización de los vídeos [752] y [753] para poder comparar el enfoque en los mismos (con filtro de Kalman extendido) con las ideas del problema planteado en el presente vídeo.

22.3.1 Identificación no lineal con filtro de Kalman extendido

[752: ekfid1] Identificación+observación estado con Filtro de Kalman extendido (1): planteamiento del problema y modelado depósitos

Materiales: [CÓD.: testIDextKF.mlx] [PDF]

**** 10:23

[YouTube ▶]

Este vídeo plantea la identificación y estimación del estado (observación) simultáneas de un sistema de depósitos con ecuaciones $\frac{dh_1}{dt} = q_e - q_{12}$, $q_{12} = b_{12}\sqrt{h_1 - h_2}$, $\frac{dh_2}{dt} = q_{12} - b_s\sqrt{h_2}$.

En este primer vídeo se plantea desde un punto de vista genérico que un parámetro constante se puede modelar con una ecuación de estado $\frac{d\theta}{dt} = 0$ o $\theta_{k+1} = \theta_k$. De hecho, esa idea es la que daba lugar a los mínimos cuadrados recursivos (vídeo [686]). Si se le añade un pequeño “ruido paramétrico” entonces tenemos el modelo $\theta_{k+1} = \theta_k + \nu_k$ que da lugar a los mínimos cuadrados con olvido (vídeo [689]) que pueden resolverse mediante un filtro de Kalman.

La idea de este vídeo es unir esos conceptos a un modelo $\dot{x} = A(\theta)x + B(\theta)u$ o, por qué no, a un modelo arbitrario $\dot{x} = f(\theta, x, u)$. Un observador para dicho sistema resolvería el problema de identificación (estimación de θ) y observación (estimación de x) simultáneamente.

La propuesta será usar el filtro de Kalman extendido, cuya teoría se discute en el vídeo [616], basada en linealización alrededor de la trayectoria observada; se remite al citado vídeo para detalles.

En la segunda parte del vídeo (a partir del instante [04:40]) se discuten los aspectos concretos del modelado del sistema de dos depósitos arriba mencionados (ecuaciones de estado y de salida), la compilación de Symbolic Toolbox a funciones numéricas con matlabFunction, y la linealización con jacobian.

La simulación de ese proceso y el filtro de Kalman extendido sobre él se abordarán en el vídeo [753].

[753: ekfid2] Identificación+observación estado con Filtro de Kalman extendido (2): simulación y resultados

Materiales: [CÓD.: testIDextKF.mlx] [PDF]

***** 17:49

[YouTube ▶]

Este vídeo desarrolla el código de simulación del problema de identificación de parámetros aproximadamente constantes y de observación del estado (de forma conjunta) planteado en el vídeo [752] sobre un sistema de dos depósitos. Por hacerlo más autocontenido, el primer minuto y medio aquí resume las ideas principales del vídeo antes referido, al que se remite para más detalle en el planteamiento del problema y el modelado.

Hasta el minuto 10 se revisa el código (bucle de simulación) con todas las linealizaciones, ecuaciones de medias y varianzas del filtro no estacionario, y simulación del proceso real.


La parte final del vídeo presenta los resultados, en tres simulaciones: una primera con los parámetros inicialmente propuestos, otra analizando el efecto de cambiar el incremento entre muestras (desv. típica) de uno de los parámetros, y finalmente se analiza el efecto de perder el sensor de altura de uno de los depósitos (varianza muy grande).

Obviamente, como alternativa “práctica” al código de este vídeo podría haberse utilizado la clase `extendedKalmanFilter` implementada en Matlab, para no necesitar “bajar” al detalle interno de las ecuaciones de filtros y linealizaciones aquí desarrolladas; el caso de estudio en los vídeos [617] y [618] se abordan de ese modo, aunque sólo hacen observación del estado de un péndulo, sin identificación.

También podría resultar de interés comparar el EKF aquí utilizado con las alternativas `unscented`/particle desarrolladas en otros materiales.

Nota: La identificación de parámetros en modelos en representación interna también se aborda en el vídeo [751], con el comando `idnlgrey`. Se aconseja la visualización del mismo para entender las similitudes y diferencias con el problema aquí planteado.

22.4 Consideraciones prácticas

[754: pdaid]	Preparación de experimentos y datos para identificación lineal en la práctica	<div>***  27:28</div> <div>Materiales: [PrepExIdent.pdf] [YouTube ▶]</div>
--------------	---	---

En este vídeo se discuten los “preparativos” que hay que hacer para tener más confianza en la utilidad de los resultados de identificación de procesos lineales ante datos reales provenientes de procesos no-lineales, de alto orden y con ruido.

Se discute la selección de una amplitud correcta de la entrada (relación señal/ruido, no-linealidad y cuantización plantean requisitos contradictorios).

Se discute también la selección del contenido en frecuencia de la entrada (ni las bajas frecuencias fácilmente controlables con acción integral, ni las altas donde el controlador no debe tener actividad son de interés): en muchos casos la entrada debería tener un contenido en frecuencias en una cierta “banda de interés”.

Los datos deberían ser de media cero, en algunos casos donde tendencias (rampa, etc.) sean claramente patentes, dicha tendencia debería ser eliminada (comando `detrend`, ver vídeo [610] para un ejemplo).

A continuación, se habla sobre la selección del período de muestreo en identificación: deben producirse incrementos “apreciables” frente al error de cuantización y al ruido y eso no ocurre si el período es demasiado pequeño.

En el experimento deben tomarse datos de ajuste y datos de validación. Los datos deben ser “limpiados” de zonas con saturaciones, outliers (muestras extrañas) o donde determinada perturbación transitoria tenga un gran efecto. Se debe también comprobar que el error de cuantización es despreciable, e invertir no-linealidades de sensor y actuador conocidas.

Por último, se discute la ventaja de filtrar entrada y salida con el mismo filtro: no cambia la función de transferencia aparente, pero el minimizar el error cuadrático sobre la salida filtrada permite enfatizar el ajuste para que sea mejor a ciertas frecuencias que a otras (mejor en aquellas frecuencias “intermedias” de interés para control y con una buena relación señal/ruido).

Capítulo 23

Sistemas dinámicos lineales en representación interna: identificación subespacio

Los métodos de *identificación subespacio* permiten identificar directamente una representación interna (modelo en variables de estado $x_{k+1} = Ax_k + Bu_k$, $y_k = Cx_k + Du_k$) de un proceso, a partir de datos experimentales de entrada-salida $\{(u_0, y_0), (u_1, y_1), \dots, (u_N, y_N)\}$ siendo tanto u como y multivariantes, si fuera el caso. También pueden identificarse, como corolario de dichas técnicas, observadores (filtros de Kalman) $\hat{x}_{k+1} = A\hat{x}_k + Bu_k + Ly_{k+1}$. Este capítulo desarrolla, a nivel introductorio, las ideas fundamentales de esta metodología.

La idea básica es utilizar las técnicas de *análisis multivariante* de la Sección 21.2 a un vector de variables construido mediante la yuxtaposición en columna de muestras de entradas y salidas de un proceso *en diferentes instantes*. Esto permite que dicho análisis multivariante identifique modelos *dinámicos* que relacionan entradas y salidas en diferentes instantes de tiempo, y que la “reducción de dimensionalidad” utilizando los componentes principales PLS o CVA pueda interpretarse como una “reducción de orden” detectando qué combinaciones de variables en diferentes instantes tienen más o menos variación.

Preliminares

[755: lsmatr] ID Subespacio 1: Mínimos cuadrados matriciales
Materiales: [IDsubespacio.pdf]


***  08:59
[YouTube ►]

Este vídeo plantea la extensión de un problema de mínimos cuadrados $y_{N \times 1} = X_{N \times M} \theta_{M \times 1}$ al caso matricial $y_{N \times Q} = X_{N \times M} \theta_{M \times Q}$. La extensión es trivial dado que se pueden considerar Q problemas de mínimos cuadrados repetidos. La fórmula pseudoinversa sigue siendo válida, así como la interpretación estadística en términos de varianzas-covarianzas.

Detalles adicionales sobre la interpretación determinista/geométrica/estadística del problema de mínimos cuadrados matriciales aparecen en el vídeo [972].

23.0.1 Ejemplo motivador: PCA de datos extendidos (lifting)

[756: stsbp] Identificación serie temporal: Análisis de componentes principales (lifting) y densidad espectral de potencia
Materiales: [Cód.: IDsubespacioserietemp.zip] [PDF]

***  07:58
[YouTube ►]

Este vídeo utiliza la serie temporal del vídeo [712]. Los dos primeros minutos iniciales del vídeo revisan en análisis PCA estático que se planteaba en dicho vídeo, mediante `svd`.


Pero, se observa que las señales y los dos primeros componentes principales no parecen "ruido blanco" en tanto a que se observa determinada relación entre las variables en diferentes instantes (componentes frecuenciales).

El vídeo compara los estimados de la densidad espectral de potencia (power spectral density) proporcionados por `periodogram` (a partir de la FFT) y `pwelch` (con cierto filtrado y suavizado de ruido), de la Signal Processing toolbox.


Representando gráficamente dichos estimados de la densidad espectral de potencia tanto de la señal en bruto como de los componentes principales, se observa que existe determinada actividad de baja frecuencia debida a dinámica.

A partir del minuto [04:45] se propone hacer *lifting* de la señal, formando un vector de muestras en diferentes instantes, para que el análisis PCA sea capaz de detectar esas relaciones entre señales en distintos instantes que definen a un sistema dinámico. Se observa que en esa matriz de datos ampliada existen cuatro o cinco combinaciones de señales en diferentes instantes que explican casi toda la variabilidad... dichas combinaciones podrían ser consideradas como "estados" de una representación interna, y la formalización correcta de estas ideas dio lugar a los métodos de subespacio de identificación de sistemas lineales en representación interna. Dicha identificación sobre esta serie temporal se aborda en el vídeo [766].

23.1 Identificación de múltiples salidas futuras (control predictivo)

[757: idmpc1]	Identificación para control predictivo: planteamiento del problema, modelo de predicción	***  11:09
	Materiales: [IDMPC.pdf]	[YouTube ►]

Este vídeo plantea el problema de genera un modelo de predicción multivariable a partir de un conjunto de entradas y salidas pasadas, así como un conjunto de entradas futuras. Se justifica la estructura de modelo basando en argumentaciones sencillas con la ecuación en diferencias de un proceso monovariable.

[758: idmpc2]	Identificación para control predictivo: ajuste por mínimos cuadrados, relación con ID subespacio, problemas subyacentes	****  10:52
	Materiales: [IDMPC.pdf]	[YouTube ►]

Este vídeo discute el ajuste por mínimos cuadrados del modelo entrada-salida para control predictivo con la estructura discutida en el vídeo [757]. Se comentan, asimismo, cuestiones derivadas de su solución no única: dado que las salidas pasadas siguen el mismo modelo que se pretende identificar existen direcciones "no controlables" que no aparecen en la matriz de varianzas de la información para la regresión; por ello, se argumenta la conveniencia de regularización o mínimos cuadrados parciales (estamos identificando un modelo de muchísimos parámetros). El SVD de las matrices resultantes de este modelo es una de las bases de los algunos de los métodos de identificación subespacio en la literatura. Dadas todas estas cuestiones de condicionamiento, efecto de ruido, etc., las ideas presentadas adolecen de defectos: este vídeo debe considerarse como material preliminar de motivación hacia las ideas de los vídeos [759] y [760].

23.2 Algoritmos de Identificación Subespacio

[759: subsp1] Identificación subespacio n4sid: conceptos preliminares

***  11:00

Materiales: [SubespacioPreliminares.pdf]

[YouTube ►]

En este vídeo se plantean los conceptos preliminares básicos para comprender las ideas principales del método de identificación subespacio: los mínimos cuadrados matriciales, para estimar la matriz Θ en una regresión $y_{m \times 1} = \Theta x_{n \times 1}$, el concepto de estado (información del pasado para predecir el futuro) y la relación de los modelos entrada-salida para predecir todas las salidas futuras hasta un cierto horizonte (discutidos en el vídeo [758]) con las matrices de observabilidad en modelos en representación interna normalizados $x_{k+1} = Ax_k + Bu_k$, $y = Cx_k + Du_k$.

[760: subsp2] Identificación subespacio n4sid: algoritmo (proyección, mínimos cuadrados parciales)

*****  23:40

Materiales: [SubespacioAlg.pdf]

[YouTube ►]

Este vídeo detalla un algoritmo de identificación subespacio basado en obtener, mediante regresión, las matrices M_1 y M_2 de un modelo de predicción $Y_{fut} = M_1 W_{past} + M_2 U_{fut}$, siendo Y_{fut} un vector de salidas futuras, U_{fut} un vector de entradas futuras, e W_{past} un vector de salidas y entradas pasadas (hasta unos horizontes futuro y pasado, h_f , h_p suficientemente largos, según lo ya discutido en vídeos [758] y [759]). La formación de las matrices necesarias para este primer paso se codifica en Matlab en el vídeo [762].

El SVD de $F = M_1 W_{past}$ permite generar una secuencia de estados no correlados entre sí que podrían explicar las salidas observadas; alternatively, blanqueando W_{past} antes de la regresión, el SVD de M_1 puede ser entendido como la solución de un problema de mínimos cuadrados parciales ortonormalizados (vídeo [722]). Si el orden del sistema es n , sólo n valores singulares de F o M_1 deberían ser distintos de cero; en situaciones prácticas, se genera la secuencia de estados haciendo cero aquéllos valores singulares muy pequeños. El código Matlab de esta etapa del algoritmo se discute en el vídeo [763].

Una vez se tiene la secuencia de estados estimados, una regresión con mínimos cuadrados matriciales, cuyo resultado utiliza una matriz pseudoinversa, como es usual, permite obtener las matrices (A, B, C, D) de la representación interna buscada, ajustando un modelo de regresión lineal:

$$\begin{pmatrix} x_{k+1} \\ y_k \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} x_k \\ u_k \end{pmatrix}$$

sustituyendo en x_{k+1} y x_k los elementos correspondientes de la secuencia de estados estimados mediante la descomposición SVD anterior. El vídeo [764] ilustra cómo ejecutar con Matlab este último paso del algoritmo de identificación subespacio.

Nota: existen bastantes variaciones del algoritmo subespacio en literatura. Por brevedad, y también porque no soy yo muy experto en ellas, no son discutidas en el vídeo aquí presentado.

[761: subspk] Identificación subespacio n4sid: identificación del observador óptimo (filtro de Kalman)

***  07:24

Materiales: [subespacioKalmanFilter.pdf]

[YouTube ►]

Este vídeo presenta una modificación sencilla a la fase final del algoritmo subespacio del vídeo [760], para, en vez de identificar un modelo (A, B, C, D) , estimar las ecuaciones de un observador actualizado (current observer) cuya ecuación de estado es $\hat{x}_{k+1} = A_{KF}\hat{x}_k + B_{KF}u_k + Ly_{k+1}$, ajustando por mínimos cuadrados matriciales el modelo de regresión lineal:

$$x_{k+1} = \begin{pmatrix} A_{KF} & B_{KF} & L \end{pmatrix} \begin{pmatrix} x_k \\ u_k \\ y_{k+1} \end{pmatrix}$$

sustituyendo en x_{k+1} y x_k los elementos correspondientes de la secuencia de estados estimados en el primer paso (SVD) del algoritmo de identificación subespacio. El ejemplo en Matlab de esta metodología para construir observadores aparece en el vídeo [765].

23.2.1 Ejemplo de implementación en Matlab

[762: sbml1] ID subespacio (ejemplo Matlab I): formación de matrices de datos

Materiales: [[CÓD.:](#) EjemploSubespacio.zip] [[PDF](#)]

*** 07:52

[YouTube ►]

Este primer vídeo de un ejemplo detallado de identificación subespacio presenta el código para generar las matrices de datos que forman el modelo de regresión básico, muy relacionado con el modelo de control predictivo:

$$\underbrace{\begin{pmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+h_f} \end{pmatrix}}_{Y_{fut,k}} = \underbrace{\bar{M}_1}_{\underbrace{\begin{pmatrix} y_{k-h_p} \\ \vdots \\ y_{k-1} \\ u_{k-h_p} \\ \vdots \\ u_{k-1} \end{pmatrix}}_{W_{past,k}}} + \underbrace{M_2}_{\underbrace{\begin{pmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+h_f} \end{pmatrix}}_{U_{fut,k}}}$$

manipulando una serie de datos. En particular, el ejemplo discute la identificación a partir de un conjunto de 3000 muestras de un sistema de 2 entradas y 2 salidas.

[763: sbml2] ID subespacio (ejemplo Matlab II): generación de secuencia de estados

Materiales: [[CÓD.:](#) EjemploSubespacio.zip] [[PDF](#)]

***** 10:41

[YouTube ►]

En este vídeo, continuación del [762], se calculan las matrices M_1 y M_2 del modelo de regresión, así como la “proyección” del futuro sobre el pasado $F = M_1 W_{past}$ –significado “formal” de proyección en vídeo [914]–. El SVD de F obtiene unos valores singulares que coinciden con los de M_1 si W_{past} se ha preblanqueado (ortonormalizado) antes de la regresión (interpretación geométrica como proyección o interpretación estadística como mínimos cuadrados parciales).

Del análisis de los valores singulares se infiere que sólo dos combinaciones lineales del vector de entradas y salidas pasadas construido en el vídeo [762] permiten predecir gran cantidad de la variabilidad del vector de salidas futuras, con lo que se estima que el orden del proceso subyacente es 2, y se genera una secuencia de estados a partir del SVD.

[764: sbml3] ID subespacio (ejemplo Matlab III): estimación de matrices (A, B, C, D) del modelo

Materiales: [[CÓD.:](#) EjemploSubespacio.zip] [[PDF](#)]

***** 07:12

[YouTube ►]

Con la “secuencia de estados estimados” construida según lo explicado en el vídeo [763], se procede a plantear un modelo de regresión para estimar las matrices del sistema (A, B, C, D) por mínimos cuadrados, en concreto:

$$\underbrace{\begin{pmatrix} \hat{x}_{k+1} & \hat{x}_{k+2} & \dots \\ y_k & y_{k+1} & \dots \end{pmatrix}}_{\Psi_{[x+,y]}} \approx \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \cdot \underbrace{\begin{pmatrix} \hat{x}_k & \hat{x}_{k+1} & \dots \\ u_k & u_{k+1} & \dots \end{pmatrix}}_{\Xi}$$

Por simplicidad, el vídeo primero explica la estimación de A y B ; tras ello, se construye el modelo de regresión para estimar C y D .

[765: sbml4] ID subespacio (ejemplo Matlab IV): filtro de Kalman, uso de ns4id *** 05:50
Materiales: [[CÓD.:](#) EjemploSubespacio.zip] [[PDF](#)] [YouTube ▶]

Este vídeo discute cómo modificar el modelo de regresión del vídeo [764] según lo indicado en el vídeo teórico [761] para estimar un observador que provea de la mejor estimación en sentido mínimo-cuadrático (filtro de Kalman) del estado x_{k+1} (extraído de la secuencia estimada en el vídeo [763]), a partir de x_k , u_k e y_{k+1} construyendo a partir de datos un observador actualizado $\hat{x}_{k+1} = A_{KF}\hat{x}_k + B_{KF}u_k + Ly_{k+1}$. El diseño basado en modelos de estos observadores por asignación de polos se aborda en el vídeo [560], y el predictor óptimo de mínima varianza en el vídeo [604]; no obstante, el algoritmo subespacio no necesita explícitamente de esa teoría para construir el observador. ****

En la segunda mitad del vídeo se comparan los resultados con los que se obtienen tras la ejecución del algoritmo `n4sid` integrado en la System Identification Toolbox de Matlab. Obviamente son muy similares. Dicho algoritmo únicamente requiere que se le introduzca la serie de datos y el orden deseado de aproximación. Si no se le introduce, lo elige él comprobando “saltos” en los valores singulares asociados; se le puede introducir un conjunto de órdenes y entonces, de forma interactiva, `n4sid(datos, ordenmin:ordenmax)` presenta los valores singulares y pide un orden al usuario para continuar.

Identificación subespacio de serie temporal

[766: stsb2] Identificación subespacio serie temporal (Matlab): n4sid, forecast, predict, residue, kalman, etfe *** 10:56
Materiales: [[CÓD.:](#) IDsubespacioserietemp.zip] [[PDF](#)] [YouTube ▶]

Este vídeo es continuación del vídeo [756], que planteaba la posibilidad de existencia de una dinámica subyacente en una serie de datos que, desde un punto de vista meramente estático se había analizado por componentes principales en el vídeo [712].

Tras una revisión de los conceptos de los referidos vídeos en el primer minuto, el vídeo, se identifica un modelo en representación interna con el comando `n4sid`, donde no existe ninguna entrada *determinista* conocida, obviamente, al ser los datos introducidos al mismo una serie temporal “aislada”.

Se compara el ajuste (FitPercent) con ordenes 1 a 6, y se considera oportuno quedarse con un modelo de orden 4... aunque a partir de orden 3 realmente ya todo es muy parecido y 3 podría ser también una buena elección.

Se comprueba con el comando `residue` que no existe autocorrelación significativa entre residuos en distintos instantes, y con el comando `xcov` que tampoco hay ninguna correlación “cruzada” que pudiera indicar dinámica no modelada importante.

Los modelos de series temporales pueden ser utilizados para predecir en bucle abierto el valor esperado de una secuencia, como aquí se ha hecho con el comando `forecast`,

también se pueden calcular observadores del estado que permitan incorporar información de medidas, aquí se usan los comandos `predict` y `kalman`.

La parte final del vídeo comprueba que el espectro en frecuencia (`spectrum`) explicado por el modelo *paramétrico* en representación interna es razonablemente cercano al calculado de forma *no paramétrica* mediante la FFT; en este vídeo, en vez de usarse `periodogram` se usa el `etfe` de la System ID toolbox, aunque son conceptualmente similares.

Capítulo 24

Detección de anomalías (ideas preliminares)

El problema de la “detección” de fallos (y de su “aislamiento”, no sólo detectar que algo no es normal, sino saber qué ha ocurrido) está muy relacionado con la identificación: básicamente se produce un “fallo” cuando el modelo que genera los datos “cambia”. El caso más sencillo sería un modelo con parámetros únicamente de “media” y “varianza”, y se debería determinar si han “cambiado”. Este capítulo aborda, de forma introductoria, algunos de los conceptos relacionados con estas ideas. El campo de la detección de fallos en sistemas dinámicos es amplio, siendo el objetivo de este capítulo únicamente el mostrar la aplicación de determinadas ideas básicas de estadística, sin mayores pretensiones: este campo no es mi especialidad ni docente ni investigadora.

Nota: El análisis estadístico riguroso con pocas muestras requiere consideraciones adicionales, como preprocesado de datos (eliminación de outliers), pruebas de normalidad (y transformaciones de los datos si no hay normalidad), pruebas de independencia, homocedasticidad (varianza condicional constante), descartar relaciones no lineales, pruebas de correlación de error, etc. Obviamente, todo ello es objetivo de cursos avanzados de Estadística y Análisis Multivariante y el objetivo de este capítulo es, simplemente, esbozar las ideas principales.

24.1 Test estadístico de hipótesis (media y varianza, mono-variable, χ^2 , t -Student)

[767: thge] Test estadístico de hipótesis: generalidades y planteamiento

Materiales: [contrastehipotesisgeneralidades.pdf]

***  13:45
[YouTube ►]

En este vídeo se discuten las generalidades sobre lo que significa el contraste o test de una cierta hipótesis. El objetivo es, dados unos ciertos datos, y una hipótesis de “funcionamiento normal”, “sin fallos”, el detectar en pocas muestras si esa hipótesis ha dejado de cumplirse. En terminología estadística, esa hipótesis se denomina $\langle \mathcal{H}_0, \text{hipótesis nula} \rangle$.

Se ponen ejemplos de hipótesis y se aclara que una hipótesis nunca podrá ser “confirmada” con un número finito de datos, pero con ellos podrá ser “invalidada” (con una cierta probabilidad de error) si los datos generados son *muy poco probables* si la hipótesis de funcionamiento normal fuera cierta.

Es necesario, por tanto, introducir niveles de confianza o significación estadística. El problema es que existe un compromiso entre la “seguridad de la detección de cambios” (nivel de confianza alto) y el retraso en confirmarlo (necesito muchas muestras), que podría ser problemático en una aplicación concreta.


Los tests más sencillos son los test de media o test de varianza (comprobar que la media o la varianza de una variable no han superado determinados umbrales).

En un caso general, el problema se complica: tomar información tiene un coste, procesarla también, retrasar la decisión a la espera de dicha nueva información podría tener consecuencias negativas (procesos inestables, enfermedades graves), y equivocarse con “falsos positivos” (diagnosticar un fallo inexistente) o “falsos negativos” (no diagnosticar un fallo existente) también. Esto es, entonces, un problema de optimización, abordado en algunos casos sencillos por resultados de lo que se conoce como *teoría de la decisión*.

Nota: Los siguientes vídeos de la sección utilizan conceptos de intervalos de confianza de distribución normal o su cuadrado. Es recomendable que el lector tenga asentados los conceptos sobre estos intervalos/elipsoides de los vídeos [intc(12:09)] y [elipc(08:22)].

[768: thmd] Test estadístico de hipótesis sobre la media

Materiales: [hipotesismedia.pdf]

***  15:43
[YouTube ►]

En este vídeo se discute en qué consiste un test de hipótesis “clásico” sobre la *media* de una variable aleatoria.

Suponiendo que la varianza del “ruido de medida” toma un valor σ conocido, si $< \mathcal{H}_0$: “la variable tiene media $\mu >$ fuera cierto”, si se toman n muestras independientes, idénticamente distribuidas, la media muestral tiende a tener una distribución $N(\mu, \frac{\sigma}{\sqrt{n}})$. Si la media muestral se aleja mucho de lo esperado (tiene muy poca probabilidad), entonces se rechaza la hipótesis.

El vídeo utiliza la función `erfcinv` para calcular los umbrales dado un cierto valor de confianza, aunque el comando `norminv` también sirve (en el PDF están ambas opciones) y, de hecho, este último es el usado en el ejemplo Matlab del vídeo [769] continuación de éste.

[09:26] Si las muestras no son independientes entre sí (correlación temporal), entonces los resultados *no* son válidos y debería usarse un filtro de Kalman (de hecho, las fórmulas en las primeras transparencias de este vídeo son un caso particular de las que resultan del filtro de Kalman, desarrolladas en el vídeo [771], cuya visualización se recomienda para darse cuenta de las similitudes con los desarrollos aquí).

[11:06] La última parte del vídeo desarrolla el caso en el que la varianza sea desconocida a priori. Entonces, como la media muestral tiene distribución normal, pero la varianza muestral es también aleatoria (tiene distribución χ^2), su cociente tiene una distribución de probabilidad que se denomina “**distribución t**”. El comando `tinva` realiza los cálculos necesarios con dicha distribución de forma sencilla.

**** ☰

[769: thmdml] Test de hipótesis sobre la media: ejemplo Matlab

Materiales: [CÓD.: testpruebamedia.mlx] [PDF]

***  10:32
[YouTube ►]

Este vídeo desarrolla un ejemplo Matlab donde se determina el número de muestras necesario para asegurar con un 99% de confianza que la media de una serie de datos es superior a 10. Para ello, se usa la teoría del vídeo [768], en concreto el caso de varianza conocida (con fórmulas de distribución normal).

Obviamente, dicho número de muestras para “detectar” el fallo depende de la media “real” de la variable medida, pero dicha media es desconocida, así como del azar, según las primeras muestras sean altas o bajas. El vídeo ilustra cómo conforme se van obteniendo más muestras, el nivel de confianza (probabilidad de acertar desmintiendo la hipótesis de “no fallo”) aumenta.

[770: thvar] Test estadístico de hipótesis sobre la varianza

Materiales: [hipotvarianza.pdf]****  10:58


[YouTube ▶]

Este vídeo desarrolla el test para invalidar una hipótesis nula que afecte a la varianza, del tipo “normalmente, la varianza es σ ” o “normalmente, la varianza debería estar por debajo σ ”, por ejemplo.

Si se han tomado n muestras independientes, se presenta el conocido test χ^2 de $(n-1)$ grados de libertad para determinar si el estimado de la varianza (varianza muestral) está lo suficientemente cerca de lo esperado (valor de referencia σ^2) o no, para, en el segundo caso, desmentir la hipótesis y detectar un “fallo”.

El vídeo presenta ejemplos numéricos bilaterales/unilaterales, usando el comando `chi2inv`.

[771: thmdkal] Ejemplo filtro Kalman: test hipótesis sobre media

Materiales: [hipotmediaKalman.pdf]****  15:25

[YouTube ▶]

Este vídeo propone un filtro de Kalman no estacionario para el sistema dinámico $m_{k+1} = m_k$ (ec. estado), $y_k = m_k + v_k$ (ec. salida). Dado que el modelo es de primer orden, sin ruido de proceso, las ecuaciones del observador (variante en el tiempo) se pueden calcular sin necesidad de utilizar Matlab.

El filtro de Kalman calcula la distribución de probabilidad *a posteriori* de m (que es la media “ideal” de y) suponiendo que es constante, tras haber tomado n muestras. Las fórmulas de distribución normal y de desviación típica que se obtienen son totalmente análogas al vídeo [768] anterior que trataba exactamente el mismo problema. No obstante, el filtro de Kalman resuelve un problema más general, que permite medias no estacionarias $m_{k+1} = m_k + w_k$ o autocorrelación temporal.

Nota: El concepto de filtro de Kalman es, en efecto, muy general y admite no sólo su uso como “observador” en sistemas dinámicos o como “estimador de la media” en contraste de hipótesis (incluso para el caso multivariable discutido en el vídeo [thmmv(12:24)]), sino también en “identificación por mínimos cuadrados”. En efecto, la identificación de parámetros “constantes” de un modelo $y = x^T \theta + v$ con x e y conocidos (siendo v un ruido de medida) es, por ejemplo, una generalización del problema de estimación de media en el vídeo [thmdkal(15:25)]: el estimador óptimo del proceso $\theta_{k+1} = \theta_k$ (parámetros constantes, esta vez entendido como una ecuación de *estado*), $y = x^T \theta$ (entendido como una ecuación de *salida*). Los vídeos [mcr1(11:18)] y [mcr2(10:51)] discuten dicha relación (mínimos cuadrados recursivos).

24.2 Test estadístico de hipótesis: caso multivariable χ^2 , T^2

Se recomienda al lector revisar el concepto de elipsoide de confianza del vídeo [elipc(08:22)] para entender mejor el umbral de decisión propuesto.

[772: thmmv] Test de hipótesis sobre media: distribución normal multivariable

Materiales: [hipotmediaMIMO.pdf]****  12:24

[YouTube ▶]

En este vídeo se discute la teoría que generaliza el test para comprobar si una media tiene o no un valor “normal” (hipótesis \mathcal{H}_0), cuyo análisis monovariable se detalló en el vídeo [768]. Tras motivar el problema multivariable y revisar el caso de una única variable (usando fórmulas de distribución normal o la χ^2_1 de un grado de libertad), se pasa al caso multivariable: determinar si existe o no evidencia para rechazar la hipótesis de que la

media de una variable aleatoria que haya dado lugar a unas ciertas muestras tiene un valor nominal μ .

Primero, se motiva las dificultades e incorrecciones de establecer un análisis “separado” para cada una de las variables: se desprecia la correlación entre ellas, de modo que cuando una supera un cierto umbral, cambian las probabilidades de que otras variables estén por encima o debajo de sus respectivos umbrales.

Tras ello, se plantea el problema conjunto. La media muestral (\bar{y} , con n muestras tomadas) de una distribución normal $N(\mu_{q \times 1}, \Sigma_{q \times q})$ se distribuye como $N(\mu_{q \times 1}, \Sigma_{q \times q}/N)$. Si $\Sigma = VDV^T$, la generalización del concepto de “desviación típica” es ahora $VD^{1/2}V^T$ y, utilizando esta última matriz en determinados cambios de variable (blaqueado), se construye un vector de variables $\eta \sim N(0, I_{q \times q})$. Si Σ es conocida a priori, $Z^2 = \|\eta\|^2 = \eta^T \eta = (\bar{y} - \mu)^T (\frac{1}{n} \Sigma)^{-1} (\bar{y} - \mu)$ se distribuye como una χ^2 de q grados de libertad, con lo que se puede generar un elipsoide de confianza en los valores de la media muestral cuyo centro sería la media nominal μ .

La parte final del vídeo discute que si la matriz Σ ha sido obtenida directamente como la varianza muestral del conjunto de valores del que también se ha extraído la media, entonces la distribución χ^2 debe cambiarse por la T^2 que es la generalización multidimensional de la t -Student introducida en la parte final del vídeo [768].

[773: thmc2m] **Test de hipótesis sobre media (varianza conocida, chi-cuadrado): ejemplo Matlab**

Materiales: [[CÓD.:](#) hipotmediaMIMOchi2.mlx] [[PDF](#)]

**** 07:37

[YouTube ▶]

Este vídeo discute un ejemplo Matlab de un test de hipótesis sobre media con varianza conocida, cuya teoría se ha discutido en la parte inicial del vídeo [772] anterior. Se plantea \mathcal{H}_0 como una matriz de varianzas-covarianzas nominal $[8 \ 4; 4 \ 10]$, y media $[5; -2]$. Se recogen 10 datos bidimensionales (que podrían formar una matriz de datos de 2×10) y se calcula su media, con el comando `mean`. Se calcula el estadístico $Z^2 = (\bar{y} - \mu)^T (\frac{1}{n} \Sigma)^{-1} (\bar{y} - \mu)$ y, usando `chi2cdf` se calcula la probabilidad de que la media muestral sea mayor o igual que Z^2 : si es baja, entonces se rechazaría la hipótesis “nominal”. Equivalentemente, con `chi2inv` se calcula un umbral de Z^2 para aceptar (si está por debajo) o rechazar (por encima) la hipótesis de que la media es $[5; -2]$.

Para comprobar el concepto de “confianza estadística”, se repiten los mismos cálculos 80000 veces. Se comprueba que siendo exactamente cierta la hipótesis nominal, en un 5% de los casos es rechazada, como la teoría predice.

Nota: Del mismo modo que el caso de varianza conocida monovariable se podía hacer equivalente a un filtro de Kalman, según discutido en el vídeo [thmdkal(15:25)], eso sigue siendo válido en el caso multivariable. Ello, por tanto, permite generalizar el test de hipótesis considerándolo como un “observador” que permitiría modelos no estacionarios con media “variante en el tiempo” $m_{k+1} = m_k + w_k$, $y_k = m_k + v_k$, siendo m un vector. Obviamente, podría generalizarse más. La idea básica es que en vez de tener un elipsoide de confianza centrado en el valor nominal μ , se tendría centrado en el valor estimado \hat{m}_k , pero los niveles también se calcularían con la distribución χ^2 . Por brevedad, los detalles se dejan al lector, dado que el desarrollo es paralelo al del vídeo [thmdkal(15:25)]. ☰

[774: thmt2m] **Test multivariable de hipótesis sobre media (varianza desconocida, test T2): ejemplo Matlab**

Materiales: [[CÓD.:](#) hipotmediaMIMOTsquared.mlx] [[PDF](#)]

***** ☰ 06:15

[YouTube ▶]

Este vídeo plantea el mismo problema que el vídeo [773], pero la varianza se supone que no es conocida *a priori*, por lo que se recogen 15 muestras y se estiman la media (\bar{y} , con el comando `mean`) y matriz de varianzas-covarianzas (\hat{S} , con el comando `cov`) muestrales. Con ello, se calcula el estadístico $T^2 = (\bar{y} - \mu)^T (\frac{1}{n} \hat{S})^{-1} (\bar{y} - \mu)$.

Como la distribución de probabilidad T^2 no está implementada en Matlab R2018a, se utiliza el hecho de que es un caso particular de la distribución F (Fisher-Snedecor) para calcular la probabilidad de que siendo cierta la hipótesis se obtengan valores de T^2 iguales o mayores al calculado por los datos. Se compara el test T^2 con el test χ^2 viendo que, debido a la incertidumbre en la varianza muestral, el test T^2 rechaza menos (umbral más alto) que el χ^2 ; ambos tests se igualan con muestras infinitas (se ha calculado con 150000 muestras en el código para comprobarlo).

24.3 Test estadísticos de detección de anomalías sobre resultados de PCA

[775: fdpca1] Umbrales de detección de cambios tras análisis PCA (1): matriz de VC completa **** 11:33

Materiales: [PCAFaultDiag1complet.pdf] [YouTube ▶]

Dada una serie de datos, el análisis de componentes principales (PCA) identifica la matriz de varianzas-covarianzas (muestral) de un proceso, y los componentes sin correlación entre sí que podrían originarla. Normalizándolos a varianza unidad, si las muestras fueran de distribución normal, también lo serían estos componentes y se podrían utilizar las fórmulas χ^2 . Con varianza desconocida, la distribución a utilizar sería la T^2 de Hottelling. De hecho, este vídeo es casi una reescritura del vídeo [772] motivándolo en el contexto del análisis de componentes principales y planteando un elipsoide de confianza como solución inicial que podría refinarse con los conceptos discutidos en los siguientes vídeos sobre esta temática ([776] y [777]).

[776: fdpcaqt1] Umbrales de detección de cambios tras análisis PCA (2): residuos dentro/fuera de modelo (motivación) *** 05:57

Materiales: [PCAFaultDiagReducido.pdf] [YouTube ▶]

En ocasiones el análisis de componentes principales es usado para considerar como “modelo” aquellos componentes con poca varianza: si los datos son n -dimensionales y existen q componentes de varianza grande, hay $n - q$ combinaciones lineales de las variables que son “aproximadamente cero”. Esto, en efecto, entronca con las ideas de los mínimos cuadrados totales (vídeo [692]): se genera un modelo de dimensión reducida asumiendo que esas combinaciones son *igual* a cero.

Ello motiva que el residuo único del vídeo [775] puede ser considerado como la contribución de dos residuos con significado diferente:

1. el residuo “en modelo” (T^2) que indica que alguno de los q componentes responsables de la variabilidad tiene un valor inusual, pero que la correlación entre variables (modelo de $n - q$ ecuaciones) no ha cambiado.
2. residuo “fuera de modelo” (Q) que indica que alguna de las $n - q$ combinaciones de variables que deberían ser cero tiene un valor significativamente diferente de cero y, por tanto, puede presumirse que el modelo que relaciona las distintas variables ha cambiado.

La motivación y explicación “informal” de estas ideas es el objetivo de este vídeo, siendo el desarrollo teórico completo abordado en el vídeo [777].

[777: fdpcaqt2]

Umbral de detección de cambios tras PCA (3): residuos dentro/fuera de modelo (detalle cálculo)

Materiales:

17:55

[YouTube ▶]

Este vídeo aborda el desarrollo teórico completo de la generación de los dos residuos (en modelo, fuera de modelo) presentados y motivados intuitivamente en el vídeo [776]. El resultado es la descomposición (T^2, Q) ampliamente usada en la literatura sobre este tema.

Otros desarrollos

La detección de fallos es un campo activo de investigación y desarrollo, que combina conceptos de estadística, control, etc. No obstante, no dispongo de más materiales realizados por mí sobre el tema dado que, como ya se ha comentado, no es mi línea principal de trabajo.


Parte V

Control Robusto


Capítulo 25

Motivación e ideas preliminares

25.1 Introducción y planteamiento del problema

[778: errmi]	El problema de control bajo errores de modelado: planteamiento general, fuentes de error	*  05:58
	Materiales: [CER.pdf]	[YouTube ►]

En este vídeo se discute la omnipresencia de errores de modelado significativo, y sus fuentes, que hacen que deban ser tomados en cuenta adecuadamente para poder diseñar controladores que funcionen de forma satisfactoria en el control de sistemas complejos en la práctica.


[779: robplant]	Control robusto: motivación y planteamiento del problema	****  14:16
	Materiales: [IntroRobcontrol.pdf]	[YouTube ►]

Este vídeo motiva y plantea brevemente los problemas a resolver en control robusto:

- los modelos son aproximaciones de las plantas reales, el control robusto utilizará modelos $G(s, \Delta)$ donde en Δ intervienen parámetros inciertos, funciones de transferencia inciertas y no-linealidades.
- La teoría lineal suponiendo que los errores de modelado son “perturbaciones” adicionales a considerar no sirve (una perturbación no puede inestabilizar un bucle nominal estable).
- Se definen los problemas de estabilidad nominal, prestaciones nominales, estabilidad robusta y prestaciones robustas.

25.2 Compromiso especificaciones-robustez

Esta sección discute en términos “prácticos” (cualitativos, sin fórmulas) qué significa el error de modelado y a qué compromisos obliga su presencia en las aplicaciones de control, se utilice o no el control robusto (que proveerá de resultados “cuantitativos”).

[780: objcr]	Objetivos del control (en general y robusto en particular)	**  07:51
	Materiales: [CER.pdf]	[YouTube ►]

Este vídeo discute los objetivos de un sistema de control: reducir variabilidad, optimizar coste, respetar restricciones ante la presencia de perturbaciones exógenas y de discrepancias entre la planta “real” y el “modelo” de la misma usado para los cálculos. Aparece la pregunta de cuál debe ser la exactitud del modelo para conseguir los citados objetivos, o de qué variables se deben medir. La pregunta es de difícil respuesta. El objetivo fundamental de un diseño “robusto” de control es que simulación se parezca a experimentación. El concepto de “prestaciones robustas” implicaría que, además de parecerse simulación a experimento, las prestaciones son aceptablemente buenas.

[781: ceridea] El compromiso especificaciones-robustez: idea básica

Materiales: [CER.pdf]

*  07:35
[YouTube ▶]

El compromiso especificaciones-robustez en control es un caso particular de una idea más general: cuanto más se optimice aprovechando “al límite” peculiaridades muy específicas de un modelo, peor será el comportamiento en caso de que el modelo sea erróneo. La idea aplica al control (calentar un edificio en 2 milisegundos se puede simular, pero nunca funcionará en la práctica) y a otros contextos como, por ejemplo, compromisos rentabilidad/riesgo en gestión de inversiones o planificación de producción.

[782: limaf] Límites de amplitud y frecuencia para tener robustez aceptable

Materiales: [CER.pdf]

***  09:23
[YouTube ▶]

Este vídeo discute las limitaciones que deberían tener las señales presentes en un bucle cerrado de control para posibilitar una robustez (probabilidad de buen funcionamiento experimental) aceptable.

Primero se discuten las limitaciones de frecuencia: los ensayos experimentales suelen ser de baja frecuencia, y aunque haya ensayos de alta frecuencia, la relación señal-ruido para identificar los modelos es peor. Por ello, las acciones de control deberían tener componentes de baja frecuencia para asegurar que la salida del proceso es razonablemente similar a la identificada.

Luego, se discuten limitaciones de amplitud: la linealización es una aproximación para amplitudes de entrada infinitesimales; mucha amplitud de entrada excita no-linealidades, pero poca amplitud dificulta la identificación basada en datos al tener una relación señal-ruido más pobre.

Nota: Los procesos inestables y de fase no mínima ven exacerbadas las limitaciones anteriores, al requerirse una amplitud mínima para estabilizar (se recomienda revisar el vídeo [minust(10:54)]) y ser muy peligrosa la saturación (en el caso inestable), o requerir una acción de control que tiende a infinito para conseguir una cierta trayectoria (caso fase no mínima). En resumen, el límite de prestaciones conseguibles (p.ej. ancho de banda de bucle cerrado) con una robustez “razonable” en procesos inestables y de fase no mínima puede ser bastante menor que en procesos “estables”-“fase mínima” con la misma amplitud del diagrama de valores singulares, según discutido en el vídeo [limprhpm(12:39)]. La saturación es uno de los errores de modelado (no-linealidad) más frecuentes en la práctica.

[783: cere1] Compromiso especificaciones-robustez: ejemplo rápido modelo primer orden

Materiales: [CER.pdf]

**  02:18
[YouTube ▶]

En este vídeo se ilustra que el modelo $G(s) = 1/(s + 1)$ que ajusta muy bien un experimento ante escalón sirve para acelerar desde 4 a 1 segundo el tiempo de establecimiento en bucle cerrado de un proceso, pero que es totalmente inútil para acelerar el tiempo de establecimiento a 0.05 segundos porque $1/(s + 1)$ describe muy mal lo que ocurren en las primeras centésimas de segundo del escalón... Todos los modelos son erróneos... dependiendo de las especificaciones buscadas $1/(s + 1)$ es un buen modelo o no lo es.

[784: cerML1]

Compromiso especificaciones-robustez: ejemplo regulador PID (Matlab)



10:14

Materiales: [[CÓD.: gd.mlx](#)] [[PDF](#)]

[YouTube ▶]

Este vídeo ilustra el compromiso especificaciones-robustez en un modelo masa-muelle-amortiguador.

En concreto, se diseña para unos parámetros nominales de un modelo de segundo orden, pero existen errores en la masa, constante de muelle y constante de amortiguador. Asimismo, se introduce dinámica de actuador rápida pero no modelada, así como un término de alta frecuencia.

Los reguladores son de tipo PID, y se diseñan dos de ellos, uno de ellos persiguiendo un comportamiento “lento” sobre el proceso nominal, y otro un comportamiento más rápido. Se comparan las simulaciones nominales de ambos diseños con simulaciones sobre los modelos de procesos con otros parámetros y dinámica no modelada, que aproximan los tipos de errores que podrían observarse en experimentos reales (que además incluirían no-linealidad y saturación seguramente).

Se observa que el diseño “lento” tiene un comportamiento más robusto (predecible, parecido a la simulación nominal) que el diseño “rápido”.

Por tanto, las consideraciones sobre especificaciones-robustez no son específicas de control multivariable o \mathcal{H}_∞ , sino que deben ser tenidas en cuenta incluso al sintonizar PIDs en modelos de primer o segundo orden.

Nota: Una vez introducidos los conceptos de “márgenes de estabilidad” para intentar evaluar la “robustez” de un diseño en la Sección 26.1, el mismo ejemplo que acabamos de ver en el vídeo [cerML1(10:14)] es estudiado en más profundidad en el vídeo [cerm3pid(11:38)]. En dicho vídeo se revelará que el diseño 1, que aparentemente es robusto aquí, adolece de inconvenientes importantes (excesiva ganancia a alta frecuencia) que hacen bastante poco aconsejable su implementación práctica sin modificaciones.

[785: cerit]

Compromiso especificaciones-robustez: enfoque iterativo en aplicaciones y discusión final



09:32

Materiales: [CER.pdf]

[YouTube ▶]


Este vídeo ahonda en los límites que tiene un modelo determinado. Todos los modelos son incorrectos: el modelo será útil mientras mejoras en las prestaciones en *simulación* se vean correspondidas por mejoras en *experimentación*. En el momento en que una mejora en simulación ya no se ve confirmada en experimentación, el modelo es inválido. Esta idea sugiere, en sistemas complejos con muchos posibles errores de modelado, el ir incrementando poco a poco las prestaciones exigidas hasta que se llegue al punto “terminal” donde el modelo ya no permita mejorar las prestaciones y deba, por tanto, identificarse un nuevo modelo.

El vídeo termina con una serie de conclusiones y reflexiones finales, que resumen las ideas principales sobre el compromiso especificaciones-robustez: deben ser tenidas en cuenta incluso para diseñar PIDs, y el *control robusto* permite dar algunos resultados *cuantitativos* sobre dicho compromiso.

25.2.1 Condicionamiento numérico

[786: OAcondnum]

Condicionamiento Numérico: Introducción y consideraciones en ingeniería de Control

*** 

11:56

Materiales: [condumshort.pdf]


[YouTube ▶]

Este vídeo presenta de forma concisa el concepto de ganancia máxima y mínima de una matriz y el número de condición (el vídeo [787] presenta los mismos conceptos con mayor extensión en tiempo). Se discute su importancia para garantizar la corrección en la práctica de soluciones de ingeniería obtenidas con modelos y datos sujetos a un cierto error, analizando la sensibilidad del resultado de unos cálculos lineales a incorrecciones en las entradas/coeficientes de matrices.

- Normas de vectores, ganancia máxima y mínima de una matriz, número de condición.
- error relativo worst-case $y + \delta y = A(x + \delta x)$
- Ejemplo numérico

[787: condnum]

Ganancias máxima y mínima de matrices, condicionamiento Numérico (versión extensa)

***  26:39


Materiales: [condum2.pdf]

[PLAY ▶]

Una versión más extensa del vídeo [786], con mayor detalle en ejemplos y en la demostración de la obtención de la ganancia máxima y mínima de una matriz A a partir de los valores propios de $A^T A$.

[788: condnml]

Inconvenientes de ganancia o condicionamiento numérico alto en ingeniería de control: ejemplo Matlab (estático)

***  15:24

Materiales: [[CÓD.:](#) condicexamlcontrol.mlx] [[PDF](#)]

[YouTube ▶]

Este vídeo presenta un ejemplo Matlab donde se trata de cancelar una perturbación medible p en un modelo $y = Gu + p$. El modelo será estático, por simplicidad (matriz de ganancia estática).

Se supondrá un modelo ya escalado, esto es, que la norma de p es menor a 1, y se desea cancelarla con una acción tal que $\|u\| \leq 1$, con lo que la ganancia mínima (mínimo valor singular) de G deberá ser mayor o igual a 1.

Se presentan tres ejemplos de G para comprobar el efecto de una ganancia o un condicionamiento excesivamente altos ante errores en posicionamiento de actuador ($u_{real} = u + \delta_u$) y de modelado ($G_{real} = G + \Delta G$):

1. un primer ejemplo con buen condicionamiento y ganancias máxima y mínima cercanas a la unidad. Aquí todo funciona razonablemente bien, los errores de modelado o de posicionamiento de actuadores se “acumulan” pero no se “amplifican” exageradamente.
2. un segundo ejemplo con buen condicionamiento pero con tanto la ganancia máxima como la mínima cercanas a 20. En ese caso, al sólo necesitar un 5% del incremento de u disponible, los errores de un 3% a “fondo de escala” en posicionamiento de u se convierten en un error relativo del 60%. Esto puede ser problemático en aplicaciones donde se desee mucha precisión en posicionamiento.
3. un tercer ejemplo mal condicionado (ganancia mínima igual a 1, ganancia máxima igual a 24). En ese caso, al realizar la maniobra difícil (de baja ganancia), un error del 3% en posicionamiento del actuador y un error de modelado del 4% acaban dando lugar a un error del 122% en el resultado final.

Los casos 2 y 3 deberían ser evitados con, por ejemplo, un rediseño del proceso, dado que posiblemente resulten “difíciles de controlar” en la práctica.

Condicionamiento óptimo. El vídeo [mincond(15:49)] demuestra cómo calcular el condicionamiento mínimo ante escalado diagonal (cambio de unidades) usando software de optimización bajo desigualdades matriciales lineales (LMI). Ese realmente sería el número de condición a considerar en una determinada planta, aunque los detalles se omiten aquí por brevedad. La metodología de ganancia relativa (RGA) usada en control descentralizado (vídeos [rga2(11:00)] y [rgacondm(07:23)]) también da cotas sobre los valores alcanzables de ese número de condición mínimo ante escalado diagonal. ****

25.3 Diseño de controladores robustos mediante optimización minimax genérica

Esta sección utiliza las herramientas de optimización genéricas, como las discutidas en la Sección 16.2, para diseñar el mejor regulador para una “familia” de procesos (de modo que las prestaciones sean aceptables incluso en el *peor* caso). Se recomienda al lector que revise las ideas de dicha sección si no está familiarizado con ellas.

[789: optrobpi1] Optimización de parámetros de modelo Simulink con incertidumbre (1): planteamiento **** 10:42
Materiales: [CÓD.: OptimizRobustaSimulinkfminconga.zip] [PDF] [YouTube ►]

En este vídeo se plantea el problema de optimizar un controlador para un proceso cuyos parámetros son inciertos. El objetivo del control óptimo para un proceso conocido es minimizar una medida de prestaciones (función objetivo, índice de coste)

$$J_{opt} := \min_{\theta, \text{params. controlador}} J(\theta)$$

En el caso de un proceso incierto, se pretenderá garantizar que incluso con el peor caso de los parámetros dentro de un cierto rango, el índice de coste está por debajo de cierto valor, esto es, resolver un problema de:

$$J_{opt} := \min_{\theta, \text{params. controlador}} \max_{\beta, \text{params. inciertos planta}} J(\theta, \beta)$$

que se conoce como optimización *minimax*. Es un problema más complejo.

Nota: El lector con una cierta familiaridad con el control óptimo, el control minimax y Simulink, podría saltarse el visionado de este vídeo y pasar directamente al vídeo [790], donde se resume en los primeros dos minutos las ideas que en este vídeo se detallan profusamente en casi once.

En el código se proponen dos casos: primero, obtener J con sólo el modelo nominal (resolviendo, por tanto, problemas como los vistos en los vídeos [585] o [592]). Como segundo caso, se presenta código que simula cinco posibles combinaciones de retardo y constante de tiempo inciertas de un proceso $G(s) = e^{-ds} \frac{1}{\tau s + 1}$. El objetivo del control óptimo minimax será minimizar el máximo coste de las cinco simulaciones (todas con el mismo regulador, claro).

En este vídeo se discute el planteamiento del problema arriba esbozado, así como el detalle del modelo de simulink y el código Matlab para resolverlos, basado en el comando `sim` que simula un modelo cuyos parámetros y variables son fijadas mediante el comando `SimulationInput`.


El índice de coste elegido es un 50% del valor de pico (máximo de la salida) + un 50% del valor de la integral del valor absoluto del error (IAE). Obviamente, los coeficientes de ponderación entre los dos componentes o añadir términos adicionales sobre pico o integral de la acción de control son variaciones de interés que se dejan al lector.

Los resultados numéricos concretos se verán en el vídeo [790] para el caso nominal, y en el [791] para el caso de planta incierta.

Nota: por simplicidad, se ha evaluado sólo el pico en el índice de coste, pero para penalizar únicamente la sobreoscilación debería programarse $\max(0, \text{pico} - 1)$. Se deja también al lector el probar esta modificación.

[790: optrobpi2] Optimización de parámetros de modelo Simulink con
incertidumbre (2): resultados caso nominal

Materiales: [[CÓD.:](#) OptimizRobustaSimulinkfminconga.zip] [[PDF](#)]

***  08:41

[YouTube ▶]

Este vídeo presenta los resultados numéricos de ejecutar el código cuyos elementos fundamentales de simulación y cálculo de índice de coste han sido descritos en detalle en el vídeo [789]. Los dos primeros minutos de este vídeo revisan de forma rápida las ideas de dicho vídeo. Dependiendo de la familiaridad del lector con estos conceptos, puede visionarse éste directamente, o después del [789], a su elección.

Una vez revisado el modelo Simulink, el código que simula una o cinco simulaciones de dicho modelo y el código que calcula el índice de coste (mezcla de IAE y de sobreoscilación), pasa a ejecutarse la optimización sobre el modelo nominal. Se busca optimizar los parámetros K_P y K_I de un regulador PI. Para que la simulación sea más rápida, se recomienda activar la opción **Fast Restart** en el modelo de Simulink.


Se comentan dos posibles alternativas (se recomienda visualizar un ejemplo sencillo mono-variable en el vídeo [927] que ilustra la misma idea):

- **fmincon** de la *Optimization Toolbox*, que realiza optimización *local* por métodos tipo gradiente, Newton, Nelder-Mead o similares (ver documentación para detalles). Puede quedarse atrapado en algún “valle” de la función de coste (mínimo local) que no sea el más “profundo” (mínimo global).
- **ga**, algoritmo genético, de la *Global Optimization Toolbox*. A cambio de un esfuerzo de cómputo bastante mayor que **fmincon** en la mayoría de los casos, consigue garantizar (con alta probabilidad si el número de simulaciones es alto) que la solución obtenida está cerca del mínimo *global*.

Con los parámetros iniciales propuestos, se comprueba que tanto **fmincon** como **ga** obtienen un resultado muy similar en este caso. Con la planta incierta y con otros parámetros de partida para la búsqueda local, eso no será así, ver vídeo [791].

[791: optrobpi3] Optimización de parámetros de modelo Simulink con
incertidumbre (3): resultados optimización robusta, mínimos
locales

Materiales: [[CÓD.:](#) OptimizRobustaSimulinkfminconga.zip] [[PDF](#)]

****  08:56

[YouTube ▶]

Este vídeo es continuación del vídeo [790], donde se resolvía un problema de control óptimo sobre un modelo primer orden + retardo con valores “nominales” de los parámetros. En este vídeo, se simulan cinco combinaciones de dichos parámetros suponiéndolos inciertos (los 4 puntos extremos dados por los máximos y mínimos de constante de tiempo y retardo + el punto central nominal).

Primero, se comprueba que el coste “peor caso” del regulador óptimo sobre el proceso nominal del vídeo [790] es, evidentemente, peor. Una de las combinaciones tiene mucha sobreoscilación: control “óptimo” no significa, en general, “robusto”... la selección de índices de coste donde el controlador óptimo sí que garantice determinada tolerancia a errores de modelado es la motivación que dio lugar a determinadas metodologías basadas en control óptimo \mathcal{H}_∞ .

Tras ello, se aborda la minimización del peor coste de las cinco simulaciones.


Se prueban varias combinaciones de parámetros iniciales. En la primera de ellas, tanto **fmincon** como **ga** obtienen el mismo resultado (mínimo *global*, la mejor solución). Sin embargo, con otras elecciones de parámetros iniciales, el comando **fmincon** se queda atrapado en mínimos locales espúreos, lejos de las prestaciones del **ga** – a cambio, es mucho más rápido en ofrecer un resultado-. Obviamente, las tolerancias, algoritmos, número de simulaciones, etc. pueden ser adecuadamente modificadas en el código para cambiarlas respecto a las opciones por defecto, pero esos detalles entran fuera de los objetivos de este material.

Capítulo 26

Análisis de estabilidad robusta

Aunque el condicionamiento numérico discutido en la Sección 25.2.1 provee de una cierta idea sobre la propagación del error al multiplicarse unos datos por una matriz o su inversa, en los problemas de control estas ideas deben generalizarse a *matrices de transferencia* y diagramas de bloques en bucle cerrado, con el objetivo de garantizar *estabilidad* pese a la posible existencia de errores de modelado. Este capítulo presenta un enfoque, basado en lo que se denomina “pequeña ganancia”, para poder abordar esos problemas en sistemas dinámicos. Por completitud, también se revisan conceptos de “márgenes” de estabilidad originados en los años 1940.

26.1 Los inicios (1940-60): márgenes de estabilidad asociados a criterio de Nyquist

[792: margn] Márgenes de estabilidad basados en criterio de Nyquist: ***  05:51
motivación y definición genérica
Materiales: [StabMargGeneric.pdf] [YouTube ►]

En este vídeo se revisa el criterio de estabilidad de Nyquist (discutido en detalle en otros vídeos). Dicho criterio, estrictamente hablando, es una condición necesaria y suficiente de “estabilidad *nominal*” de un bucle cerrado $S(s) := 1/(1 + L(s))$ según las vueltas que la respuesta en frecuencia de bucle abierto $L(j\omega)$ da al punto -1 en el plano complejo.

Ya en los años 1940 se advirtió que la “distancia” al punto -1 podría ser indicativa de la tolerancia a errores de modelado (estabilidad “*robusta*”) de un cierto diseño. En este vídeo, por tanto, se define un margen de estabilidad genérico de $L(s, \delta)$ como la máxima modificación admisible m_δ de un cierto parámetro δ incierto hasta que el diagrama de Nyquist “toca” al -1 , esto es, hasta que exista una frecuencia ω^* tal que $L(j\omega^*, m_\delta) = -1$. Los parámetros δ más usados son la ganancia, desfase y retardo, desarrollados en el vídeo [793].

[793: margfg] Márgenes de fase, ganancia y retardo en control monovariable ***  07:09
Materiales: [StabMargFaseGanancia.pdf] [YouTube ►]

Este vídeo discute el significado de los márgenes de fase, ganancia y retardo para determinar la estabilidad robusta de un sistema monovariable en bucle cerrado. El margen de ganancia m_g es el incremento (o decremento, si el sistema fuera inestable en bucle abierto) $L' = m_g L$ que antes inestabiliza el bucle, el margen de fase ϕ es el máximo ángulo tal que $L' = e^{-j\phi} L$ toca el límite de estabilidad, y el margen de retardo d es aquél valor que hace que $L' = e^{-ds} L$ toque el límite de estabilidad.

[794: margfgm] Márgenes de fase, ganancia y retardo (control monovariable): ejemplo Matlab *** 09:33

Materiales: [Cód.: ganfasemargen.mlx] [PDF] [YouTube ▶]

Este vídeo presenta un ejemplo en Matlab de cómputo de los márgenes de fase, ganancia y retardo cuya definición teórica se ha discutido en el vídeo [793]. Los comandos relacionados son `allmargin` y `margin`.

Diseño de compensadores de adelanto-atraso. Las técnicas de diseño de reguladores que garantizan unas ciertas prestaciones (ancho de banda, tiempo de subida) y robustez (márgenes de fase y ganancia) suelen enmarcarse bajo la denominación de diseño de compensadores de adelanto-atraso. Básicamente, el compensador de adelanto es similar a la acción PD (con filtro de ruido), y el compensador de atraso es similar a la acción PI. Son técnicas básicamente gráficas para sistemas de una entrada y una salida, originadas a mitad del siglo XX que, por brevedad, no son discutidas aquí. Se remite al lector interesado a los libros de texto adecuados. Históricamente, resultan de relevancia, porque son la primera aproximación más o menos acertada al diseño de controladores robustos. Estas técnicas perseguían obtener controladores de un ancho de banda “razonable”, con márgenes de fase mayores de 45° , y márgenes de ganancia mayores de 2 (6 dB)... otras recomendaciones algo más exigentes indicarían intentar obtener márgenes de fase mayores de 60° , y márgenes de ganancia mayores de 4 (12 dB)... Usadas “sensatamente” en procesos sencillos dan buenos resultados, pero tienen ciertos inconvenientes, cuyo análisis es objetivo de la Sección 26.1.1. *** ☰

26.1.1 Inconvenientes de los márgenes de fase/ganancia/retardo

[795: mgfgsi] Margen de fase y ganancia (Nyquist): robustez a desviaciones simultáneas de fase y ganancia (Matlab) *** ☰ 08:39

Materiales: [Cód.: ganfaseErrorSimult.mlx] [PDF] [YouTube ▶]

En este vídeo se presenta un ejemplo Matlab de un sistema con una resonancia a frecuencias próximas al ancho de banda de bucle cerrado. En ese caso, existe un margen grande de tolerancia a las perturbaciones en ganancia y fase, pero de forma *separada* (manteniendo el otro parámetro de incertidumbre a cero). Sin embargo, una modificación *simultánea* de la ganancia y el retardo de solamente un 8% del margen de cada una de ellas es capaz de inestabilizar al sistema cuando se cierra el bucle.

[796: mgfginc] Margen de fase y ganancia (Nyquist): inconvenientes metodológicos **** 10:47

Materiales: [Cód.: ganfaseinconvenientes.mlx] [PDF] [YouTube ▶]

En este vídeo se revisan los inconvenientes metodológicos de las técnicas basadas en margen de fase y ganancia:

- Los márgenes miden robustez ante perturbaciones de ganancia y fase aisladas, pero podrían pecar de optimistas ante modificaciones simultáneas (resume brevemente la idea explicada en detalle en el vídeo [795]).
- La distancia al -1 indica la tolerancia a cambios en $L = GK$... pero para un mismo L , la tolerancia a errores de modelado $G_{real} = G_{modelo} + \Delta$ depende del regulador K , porque $L_{real} = GK + K\Delta$. En definitiva, un análisis de prestaciones y robustez examinando únicamente L se deja en el tintero determinadas características importantes... en particular, la posible amplificación de ruidos de alta frecuencia en la acción de control de $K/(1+GK)$ que excitarían saturaciones, romperían elementos mecánicos por fatiga y, como se verá en el vídeo [799], también tiene interpretación de tolerancia (o falta de ella) ante ciertos errores de modelado.

- La metodología gráfica que era “atractiva” antes de tener computadores personales, resulta “tediosa” (hay que mirar muchas gráficas a la vez, hay requisitos contradictorios) a la hora de sintonizar reguladores para procesos complejos, al menos comparada con las técnicas de diseño basadas en optimización \mathcal{H}_∞ , discutidas en otros materiales sobre el enfoque moderno al control robusto.
- Si ya los inconvenientes anteriores se producen incluso en casos monovariantes, el caso multivariable resulta, esencialmente, inabordable (existen muchos errores de ganancia y fase simultáneos en cada elemento de una matriz de transferencia $G(s)$, los reguladores son también matrices $K(s)$).

26.1.2 Caso de estudio

Aparte de la sección anterior, se recomienda al lector revisar el vídeo [rfbc(16:46)] sobre la evaluación de prestaciones de bucle cerrado en el dominio de la frecuencia.

[797: cerm3pid] **Caso de estudio: análisis comparativo en frecuencia de prestaciones y márgenes de estabilidad de tres PID's**

Materiales: [[Cód.:](#) cerml1gdmargenesNyquist.mlx] [[PDF](#)]

*** 11:38

[YouTube ▶]

En este vídeo se analiza si los márgenes de fase, ganancia y retardo son capaces de predecir la tolerancia a error de modelado de tres posibles PID's para un proceso masa-muelle-amortiguador, sin la simulación “exhaustiva” del vídeo [784] (repetida aquí por conveniencia: el proceso y los reguladores 1 y 2 son los mismos que en dicho vídeo, para poder analizar todo de forma conjunta).

La respuesta es afirmativa en el caso del diseño 2, excesivamente rápido y que tiene un margen de retardo demasiado pequeño (aunque no así de “fase”, aparentemente correcto). Sin embargo, los márgenes de ganancia, retardo y fase del diseño 1 son correctos según las indicaciones de muchas técnicas de diseño en frecuencia, muy parecidos al diseño 3. No obstante, la actividad de alta frecuencia de $KS = K/(1 + GK)$ es muy alta tanto en los diseños 1 como 2, por lo que seguramente no funcionarían en la práctica como se esperaría de las simulaciones nominales (KS es la función de transferencia entre perturbaciones a la salida y acción de control, con lo que amplificarían demasiado el ruido de medida de alta frecuencia). Sólo el diseño 3 tiene buenos márgenes de fase, ganancia y retardo y poca actividad de la acción de control (KS) a alta frecuencia. La función KS , a parte de implicar una degradación de prestaciones en cuanto a amplificación de ruido de medida sobre los actuadores, tiene un significado de *robustez* explícito, discutido en el vídeo [799].

Nota: La comparación de las conclusiones de este vídeo con los márgenes de robustez basados en pequeña ganancia (en los que se basa el diseño \mathcal{H}_∞ moderno, cuyo estudio se realiza en otros materiales a continuación) es estudiada en el vídeo [810].

26.2 Teorema de pequeña ganancia

[798: OApeqga] **Teorema pequeña ganancia**

Materiales: [PGOA.pdf]

*** 10:58

[YouTube ▶]

Este vídeo describe el teorema de pequeña ganancia.

- Motivación: de la incertidumbre muchas veces sólo se conoce una “cota”
- Revisión de norma 2, desigualdad triangular y norma infinito de un sistema (detalles en vídeo [633]).

- Desigualdad submultiplicativa en norma ∞ .
- Estabilidad BIBO: definición equivalente como finitud de norma ∞ .
- Enunciado del teorema de pequeña ganancia. Demostración.
- Conclusiones

Nota: otra forma complementaria de probar estabilidad en control lineal y no lineal es el concepto de “pasividad” que es la abstracción de la “disipación de energía” de los sistemas físicos; este concepto se discute en los vídeos [878] y [879].


Nota: Los márgenes de pequeña ganancia son más conservativos que los márgenes que podría dar el criterio de Nyquist... a cambio, se aplica a una clase de sistemas mucho más amplia: puede garantizar estabilidad ante no-linealidades o variación en el tiempo de los sistemas interconectados.

26.3 Análisis de estabilidad robusta mediante pequeña ganancia

26.3.1 Incertidumbre aditiva

[799: IncAdd] Análisis de robustez ante Incertidumbre aditiva: margen de estabilidad

Materiales: [IncertAd.pdf]

***  10:59

[YouTube ►]

Este video analiza la estabilidad de $G_{real} = G + \Delta_+$ (incertidumbre aditiva) en bucle cerrado ante un controlador K si se sabe que el bucle cerrado entre G y K (modelos conocidos) es estable y se tiene únicamente una cota de $\|\Delta_+\|_\infty$ (incertidumbre acotada no estructurada).

Se utiliza el teorema de pequeña ganancia.


También se enuncia el teorema de pequeña ganancia en versión frecuencia a frecuencia, que permite ser menos conservativo si se sabe que G_{real} es lineal invariante en el tiempo y se tiene un cota en frecuencia de Δ_+ , $\|\Delta_+(j\omega)\| \leq \|W(j\omega)\|$.

Se esboza la idea de calcular una cota de $\|\Delta_+\|_\infty$ basado en simulación exhaustiva, variando parámetros, retardos, etc.

Ejemplos Matlab de uso de estas fórmulas se desarrollan en los vídeos [800] (monovariable) y [801] (multivariable).

[800: IncAddML] Análisis margen de estabilidad ante Incertidumbre aditiva: ejemplo Matlab

Materiales: [IAML.pdf]

***  10:59

[YouTube ►]


En este vídeo se aplican los resultados sobre margen de estabilidad del vídeo anterior [799] sobre el sistema masa-muelle-amortiguador controlado con PIDs en el que se observó el significado “intuitivo” del compromiso especificaciones-robustez en el vídeo [784].

- Mediante las fórmulas de pequeña ganancia, se consigue probar que el regulador “rápido” es 20 veces más sensible a error de modelado que el regulador “lento”.
- Variando aleatoriamente los parámetros de masa, constante elástica, constante de amortiguación y ancho de banda de actuador, se consigue estimar un filtro $W(s)$ que acota la incertidumbre aditiva de la familia de modelos.
- Las fórmulas de pequeña ganancia demuestran, antes de simular/experimentar, que el regulador “lento” es estable para toda la familia de modelos acotada, mientras que el “rápido” no lo es.

Un caso multivariable de este tipo de análisis se estudia en el vídeo [801].

[801: IncAd2]

Análisis margen estabilidad ante Incertidumbre aditiva: ejemplo
Matlab multivariable

****  09:56

Materiales: [[CÓD.:](#) AnalisIncertA2.mlx] [[PDF](#)]

[YouTube ▶]


En este vídeo se aplican los resultados teóricos sobre margen de estabilidad del vídeo [799] sobre un sistema de dos entradas y dos salidas.

El desarrollo es similar al caso monovariable del vídeo [800]:

- Variando aleatoriamente los parámetros de ganancias y constantes de tiempo se genera una familia de modelos cuya estabilidad en bucle cerrado se desea garantizar.
- las fórmulas de pequeña ganancia, dado que Δ es de dimensión 2×2 acotan la ganancia máxima de Δ en función de la ganancia mínima de $\text{inv}(K \cdot (I + GK)^{-1})$, que coincide con la inversa de la ganancia máxima de $K \cdot (I + GK)^{-1}$; ver propiedades SVD en vídeo [933].
- Se representa gráficamente tanto la cota de la norma infinito (válida para incertidumbre no lineal o variante en el tiempo) o la cota frecuencia a frecuencia (válida para incertidumbre lineal invariante en el tiempo, como es el caso de la familia de plantas generada).
- Se comparan dos reguladores, obteniendo los de ganancia menor un mayor margen de robustez (que es lo intuitivamente razonable para plantas *estables* en bucle abierto).
- Se observa el conservadurismo del teorema de pequeña ganancia: incluso si el umbral de error es rebasado, todos los bucles cerrados son estables y con unas prestaciones muy parecidas. Ello motiva el estudio en profundidad del problema de prestaciones robustas y teorema de pequeña ganancia *escalado* (vídeo [835]), aunque una primera aproximación basada en una sencilla modificación de la fórmula de pequeña ganancia es abordada en el vídeo [802] sobre el mismo proceso que en este ejemplo (con esa nueva modificación, ya se consigue probar estabilidad robusta).

[802: IncAd2s]

Ideas preliminares análisis margen estabilidad ante
Incertidumbre aditiva estructurada (sólo direccionalidad
MIMO): ejemplo Matlab 2x2

****  08:45

Materiales: [[CÓD.:](#) AnalisIncertA2estruprelim.mlx] [[PDF](#)]

[YouTube ▶]


Este vídeo discute la idea que la incertidumbre no tiene por qué ser "esférica" y, por otro lado, en un caso multivariable las matrices que multiplicadas por Δ deben dar menor a 1 tienen diferente ganancia en diferentes direcciones, por lo que pueden tolerar más error en unas direcciones que en otras.

En este vídeo se discute la idea (para el caso aditivo, aunque valdría para otros) de cambiar el teorema de pequeña ganancia $\|\Delta\|_\infty \cdot \|K(I + GK)^{-1}\|_\infty \leq 1$ por la condición $\|\Delta(j\omega)\| \cdot \|K(I + GK)^{-1}\|_\infty \leq 1$, esto es *frecuencia a frecuencia*. La segunda es menos conservativa, ya que Δ puede ser grande a frecuencias donde $K(I + GK)^{-1}$ es pequeño. A cambio, la incertidumbre tiene que ser lineal e invariante en el tiempo para que exista $\Delta(j\omega)$. La idea puede argumentarse del criterio de Nyquist, evitar que ningún punto del eje imaginario sea raíz de la ecuación característica de bucle cerrado $\det(I + \Delta(s) \cdot K(s) \cdot (I + G(s)K(s))^{-1}) = 0$.

En el ejemplo concreto, se comprueba que aunque las cotas en frecuencia planteadas en un primer momento (vídeo [801]) se violaban, con esta nueva fórmula ya no se superan dichos umbrales y por tanto puede garantizarse estabilidad robusta. La reformulación con más rigor y generalidad de estas ideas dio lugar al análisis de *valor singular estructurado*, que no entra en los objetivos de este material.

[803: ucover1]

Ucover (Matlab): estudio de su funcionamiento y
conservadurismo (SISO)

****  08:19

Materiales: [[CÓD.:](#) pruebasUcoverSISO.mlx] [[PDF](#)]

[YouTube ▶]

En este vídeo se genera una familia de plantas modificando los parámetros y añadiendo dinámica de alta frecuencia y retardo a un modelo nominal de segundo orden (SISO).

Se usan los comandos `stack`, `frd` y `ucover` para obtener una representación de la incertidumbre $G + \Delta \cdot w_1(s)$, siendo $w_1(s)$ un filtro (escalar) de orden 2.

Este tipo de “recubrimiento” tiene ventajas computacionales en cálculo de márgenes de estabilidad y reguladores, pero a cambio se es conservativo en ciertos aspectos, que se analizan aquí.

Se representan, para distintas frecuencias, en el plano complejo (diagrama de Nyquist):

- El círculo de incertidumbre dado por $|w_1(j\omega)|$
- El centro del círculo $G(j\omega)$
- La nube de puntos $G_{real,i}(j\omega)$, $i = 1, \dots, 50$
- El círculo más pequeño que incluiría a la nube de puntos anterior.

Se analizan las tres causas de conservadurismo:


1. El modelo nominal puede no estar en el “centro de gravedad” de la nube de puntos,
2. La incertidumbre estructurada no se distribuye uniformemente en todo el círculo de incertidumbre *no* estructurada.
3. El orden pequeño de $w_1(s)$ sobreacota significativamente la aproximación en bastantes puntos de frecuencia.

Por contra, puede que no se hayan considerado no-linealidades o el “realmente peor caso” en la familia de plantas al azar considerada, con lo que en esos aspectos se es “arriesgado”.

26.3.2 Otras incertidumbres no estructuradas (multiplicativa, coprime, ...)

[804: incotra] Incertidumbre no estructurada (2): incertidumbres multiplicativas, inversas, etc.

Materiales: [IncertMultOtra.pdf]

****  21:36


[PLAY ►]

Este vídeo es continuación de la teoría expuesta en el vídeo [799]. Aparte de una revisión de conceptos y motivación, a partir del minuto [02:30] se presentan las fórmulas para incertidumbre no estructurada multiplicativa en entrada $G_{real} = G(I + \Delta)$, multiplicativa en salida $G_{real} = (I + \Delta)G$, aditiva inversa $G_{real} = (G^{-1} + \Delta)^{-1}$, multiplicativa inversa en salida $G_{real} = (I + \Delta)^{-1}G$, y, por último, multiplicativa inversa en entrada $G_{real} = G(I + \Delta)^{-1}$.

Se presenta una tabla resumen con todas las fórmulas, incluyendo pesos/escalados en Δ , esto es cuando la incertidumbre en un proceso es $\hat{\Delta}$, siendo $\hat{\Delta} = W_1 \Delta W_2$ con $\|\Delta\|_\infty < 1$. Obviamente, los pesos W_1 y W_2 pasarán a la parte “determinista” conocida de la planta para aplicar pequeña ganancia.

[805: incf1] Incertidumbre no estructurada en representación factorizada (coprime factor uncertainty)

Materiales: [IncertCop1.pdf]

****  09:33

[YouTube ►]

En este vídeo se presenta el análisis mediante de estabilidad robusta fórmulas de pequeña ganancia de un proceso cuyo modelo nominal es $P = M^{-1}N$, pero cuyo modelo incierto es $P(\Delta) := (M + \Delta_M)^{-1}(N + \Delta_N)$. Esta forma de incertidumbre resulta conveniente para errores en “numerador” y “denominador” particularmente si el modelo es inestable


(dado que la representación “directa” usando otras posibilidades de entre las discutidas en el vídeo [804] podrían resultar en un Δ inestable o no realizable).

Como “curiosidad” (opcional, no relacionada con los conceptos de robustez aquí planteados), las representaciones factorizadas se originaron históricamente para plantear generalizaciones del control IMC, por modelo interno (ver vídeo [420]).

La interpretación de la incertidumbre en los factores coprimos relacionada con otras incertidumbres se aborda en el vídeo [806]. También en los vídeos [807] y [808] se aborda cómo obtener la “distancia” en incertidumbre en factores coprimos normalizados entre dos plantas y su interpretación en la respuesta en frecuencia.

[806: incf2] Incertidumbre en representación factorizada: ampliación

Materiales: [IncertCop2.pdf]

*****  20:21

[YouTube ▶]

En este vídeo se amplían conceptos sobre la representación factorizada $P = M^{-1}N$ introducida en el vídeo [805]. En concreto, como dicha factorización no es única, se define una factorización “normalizada” donde M y N son estables, primos entre sí y están normalizados a 1 en la respuesta en frecuencia ($MM^H + NN^H = I$).

Se presentan comandos de Matlab para calcularla, y también se introduce la reducción de orden en factorización coprime normalizada (`ncfmr`), que permite obtener una factorización M_r, N_r $P_r = M_r^{-1}N_r$ de modo que los errores Δ_M y Δ_N sean suficientemente pequeños (se basa en la reducción equilibrada de M y N).

Dado que la factorización normalizada tiene respuesta 1 a todas las frecuencias, no cambia la norma infinito si se multiplica una matriz de transferencia por $Q = [M \ N]$, con ello se presenta una fórmula de cálculo del margen de estabilidad de un regulador K que sólo depende de P y de K , por lo que el cálculo explícito de M y N normalizados no es necesario. En efecto, se demuestra que:

$$\begin{aligned} \|T\| &= \left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - PK)^{-1} M^{-1} \right\| = \left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - PK)^{-1} M^{-1} (M \ N) \right\| \\ &= \left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - PK)^{-1} (I \ P) \right\| := \|\tilde{T}\| \end{aligned}$$

siendo T la fórmula obtenida en el vídeo [805], pero \tilde{T} tiene mucha más facilidad de ser obtenida en la práctica, y una interpretación (como se verá a continuación) también interesante. **Nota:** como su norma es la misma, el vídeo intercambia T y \tilde{T} libremente usando una única notación T refiriéndose a una u otra según el contexto; las transparencias, no obstante, han sido corregidas para distinguirlas y evitar posibles confusiones.

El comando de Matlab que calcula dicho margen es `ncfmargin`.

La última idea que se presenta en el vídeo es la relación de la “incertidumbre aditiva en factorización normalizada” con el “error porcentual en denominador y numerador” y, en general, con una combinación de incertidumbres “aditiva”, “aditiva inversa”, “multiplicativa” y “multiplicativa inversa” sobre la planta no factorizada (descritas en el vídeo [804]): el margen ante incertidumbre coprime normalizada es un margen de robustez ante “un poco de error por todas partes” entre todas las entradas y salidas al controlador, lo cual da una interpretación “práctica” interesante a este tipo de incertidumbre: si ese margen es suficientemente grande ninguna de las funciones de transferencia entre cualquier entrada o salida tendrá ningún “pico indeseable”.

Nota: El vídeo aquí resumido solamente contempla el *análisis* de estabilidad con un controlador prediseñado (por cualquier técnica); el controlador que optimiza ese margen de incertidumbre se discute en el vídeo [824], pero suele dar prestaciones muy poco agresivas (en un proceso estable, lo mejor para no inestabilizar es “no tocar nada”), la (casi siempre necesaria) incorporación de pesos para mejorar prestaciones se discute en el vídeo [825].

La distancia entre dos plantas en términos de incertidumbre coprime normalizada se conoce como ν -gap, que calcula Matlab en el comando `gapmetric`. Una discusión sobre la interpretación de esta distancia se aborda en los vídeos [807] y [808].

Consideraciones adicionales incertidumbre factores coprimos y nu-gap metric

[807: gapm1]

incertidumbre factorización normalizada, métrica nu-gap:
interpretación geométrica simplificada (real, SISO)

Materiales: [nugapmetricrealSpanish.pdf]

📺 15:49

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute la interpretación geométrica de la incertidumbre en la factorización coprime normalizada, introducida en los vídeos [805] e [806], asociada a los comandos de Matlab `ncfsyn`, `ncfmargin`, `gapmetric`.

En concreto, se plantea cómo traducir la distancia entre dos plantas P_1 y P_2 a cambios en numerador y denominador normalizados. Por simplicidad, se presenta la idea en el caso de que P_1 y P_2 sean simples números reales, para una comprensión más intuitiva.

La idea que se desarrolla es que la fracción $p = n/d$, representada por un par (d, n) puede ser entendida por la recta $q \cdot (d, n)$ porque (qd, qn) representa también a $p = (qn)/(qd)$. Entonces p es la pendiente de dicha recta.

La mínima desviación en la representación normalizada (denominada *nu-gap*) que transforma $p_1 \equiv (d_1, n_1)$ en $p_2 \equiv (d_2, n_2)$ es el *seno* de la diferencia entre ángulos de la recta asociada a cada p_i .

En la segunda mitad del vídeo se discuten fórmulas para calcular ese *nu-gap* entre dos números p_1 y p_2 , tanto a partir de numerador y denominador normalizados como a partir de los valores originales.

Por último, se introduce una segunda interpretación como distancia cordal sobre la proyección estereográfica, que es la propuesta en los trabajos seminales de G. Vinnicombe al respecto.

Aunque esta interpretación geométrica es generalizada en los citados trabajos al caso complejo multivariable, no se aborda aquí por brevedad y simplicidad. La discusión sobre esta incertidumbre continua en el vídeo [808] de esta colección.

[808: gapm2]

incertidumbre factorización normalizada: métrica nu-gap, caso
general (sin demostraciones)

Materiales: [nugapmetric2ESP.pdf]

📺 09:44

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo continua la discusión del vídeo [807], que revisaba la interpretación geométrica del ν -gap en el caso real. Aquí se generaliza al caso complejo de distancia ν -gap entre $P_1(j\omega) \in \mathbb{C}$ y $P_2(j\omega) \in \mathbb{C}$, también interpretable en el caso SISO como distancia cordal en proyección estereográfica. Las fórmulas para matrices de transferencia MIMO se enuncian sin demostración y se comentan brevemente. Este tipo de interpretación está detrás de los cálculos realizados en el comando `gapmetric` de la Robust Control Toolbox de Matlab.

Si, dado un regulador K , el margen `ncfmargin` ante incertidumbre en factorización coprime normalizada es mayor que el *nu-gap* entre la planta nominal y la real, entonces la planta real será estabilizada por K .

[809: ncfcsc]

Efecto del escalado en el margen ante incertidumbre en
factorización coprime normalizada

Materiales: [ncfmarginescalado.pdf]

📺 12:18

[YouTube ▶]

Este vídeo discute la influencia del escalado en los valores de `ncfmargin(planta,controlador)`. En efecto, el escalado cambia la factorización coprime normalizada o, en la segunda interpretación (margen de robustez ante una combinación de cuatro incertidumbres) escala las incertidumbres aditivas de forma directamente proporcional al factor de escala que se aplique a un modelo, y las aditivas inversas de forma inversamente proporcional. Las multiplicativas quedarían inalteradas en el caso monovariable. En el caso multivariable con escalados diferentes para cada entrada y salida el resultado es más complicado. Como resultado, pese a significar el mismo bucle de control, `nfcmargin` devuelve un resultado diferente con `ncfmargin(P,K)` y con `ncfmargin(P/100,K*100)` simplemente por cambiar de centímetros a metros una salida.


Por ello, se propone escalar a ganancia cercana a uno dado que el 10% de la unidad es comparable al 10% de incertidumbre multiplicativa, y al 10% de la inversa (la inversa de 1 es 1). En otros casos, la interpretación del valor de `ncfmargin` resulta más dudosa y, quizás, es mejor analizar por separado cada uno de sus cuatro componentes. Un ejemplo Matlab de estas cuestiones se aborda en el vídeo [823].

26.3.3 Caso de Estudio

[810: `cermpg3pid`] Caso de estudio: márgenes de estabilidad (pequeña ganancia vs. Nyquist) de tres PID's

10:49

Materiales: [[CÓD.:](#) `cerml1gdmargenessmallgain.mlx`] [[PDF](#)]

**** 

[YouTube ▶]

Este vídeo continua el caso de estudio de analizar formalmente el compromiso especificaciones-robustez de tres PID's, discutido a base de simulación exhaustiva en el vídeo [784], pero dado que simular el “peor caso” es básicamente como “encontrar una aguja en un pajar”, motivó el análisis de los márgenes de estabilidad “clásicos” (fase, ganancia, retardo) en el vídeo [797].

Aquí se revisan muy brevemente las conclusiones de dicho vídeo (hasta el minuto 02:45), y luego se computan los márgenes basados en pequeña ganancia considerando incertidumbre aditiva ($1/\|K/(1+GK)\|_\infty$), multiplicativa ($1/\|KG/(1+GK)\|_\infty$), aditiva inversa ($1/\|G/(1+GK)\|_\infty$) y multiplicativa inversa ($1/\|1/(1+GK)\|_\infty$). La tolerancia ante una combinación arbitraria de esas cuatro incertidumbres es, de hecho, el margen de robustez ante incertidumbre coprime normalizada no estructurada, evaluado con el comando Matlab `ncfmargin`, según discutido en el vídeo [806] (de hecho, el lector falto de tiempo puede saltar dicho vídeo teórico y concentrarse en el significado “práctico” y uso en Matlab discutidos aquí). Dicho margen es también calculado y comparado con los otros resultados de pequeña ganancia.

Nota: Este caso de estudio sólo ha abordado un *análisis* de estabilidad robusta; el desarrollo es continuado, incorporando metodologías de **diseño** de reguladores robustos “mixed sensitivity” y “Glover-McFarlane loop shaping” en los vídeos [c3pidmx(13:01)] y [c3pidncf(10:54)], respectivamente, una vez las técnicas de diseño mencionadas hayan sido correctamente presentadas en los distintos vídeos del Capítulo 27 a continuación.

Capítulo 27


Diseño de controladores robustos ante incertidumbre no estructurada

Una vez visto en el capítulo 26 cómo analizar si un controlador *preexistente*, diseñado por cualquier metodología, puede tolerar unos ciertos errores de modelado sin hacerse inestable, este capítulo aborda el cómo utilizar las técnicas \mathcal{H}_∞ para *diseñar* controladores robustos (intentando, claro, mantener unas ciertas prestaciones), aprovechando los conceptos de planta generalizada y control óptimos vistos en el capítulo 19. La metodología \mathcal{H}_∞ es razonablemente fiable desde el punto de vista numérico y genera rápidamente controladores multivariables que solucionan este tipo de problemas. En el capítulo 28 se planteará un tipo de incertidumbre más general (estructurada, varias fuentes de incertidumbre Δ) y técnicas iterativas para solucionarlo; no obstante, esas nuevas técnicas son computacionalmente mucho más costosas que las que aquí se proponen y, en bastantes ocasiones, pueden fallar. También pueden minimizarse índices de costes temporales sujetos a restricciones de picos de respuesta en frecuencia con las herramientas de Simulink Response Optimization (vídeo [sdofreq(09:26)]).

27.1 Sensibilidad mixta

La metodología de “sensibilidad mixta” (mixed sensitivity) es una metodología de *diseño de reguladores* para dar una forma adecuada a las funciones de transferencia (o matrices, en el caso multivariable) de bucle cerrado discutidas en el vídeo [rfbc(16:46)], así como en el caso de estudio Matlab [rfbcm1(14:22)], cuyos contenidos se recomienda revisar para una mejor comprensión de los conceptos de esta sección.

De hecho, ya desde el punto de vista de control óptimo \mathcal{H}_2 o \mathcal{H}_∞ , sin tener consideraciones de robustez, como se van a incluir aquí, el problema de sensibilidad mixta tenía una interpretación bastante clara e interesante, como se discutió en el vídeo [ef1(16:26)]. Lo que se hará en esta sección es comprobar que esas ideas tienen también una interpretación (dada por el teorema de pequeña ganancia) interesante en control robusto (tolerancia a errores de modelado).

[811:mxs1] **Sensibilidad mixta / mixed sensitivity (1): motivación e ideas preliminares** *******  **10:58**

Materiales: [mxs.pdf] [YouTube ▶]

Este video revisa conceptos y describe las ideas preliminares del método de sensibilidad mixta (*mixed sensitivity* en inglés), en controladores de un grado de libertad.

- Se revisa el teorema de pequeña ganancia y la fórmula del margen de estabilidad ante incertidumbre aditiva y multiplicativa. Detalles en los vídeos [??] y [804].
- Se obtienen las matrices de transferencia para el bucle estándar realimentado de 1 grado de libertad entre referencia y error (sensibilidad $S = (I + GK)^{-1}$), acción de control (sensibilidad de control $KS = J(1+GK)^{-1}$), y salida (sensibilidad complementaria $T = (I + GK)^{-1}GK$).

- Las dos últimas coinciden con las fórmulas del margen de estabilidad aditiva/multiplicativa.
- Basado en la idea anterior, se plantean conclusiones “intuitivas” de cara a diseño robusto: limitar resonancia en bucle cerrado (sobreoscilación) y limitar amplitud de control. Compromiso especificaciones-robustez.

Transformar todas estas ideas en una metodología de síntesis de reguladores mediante control óptimo \mathcal{H}_∞ se aborda en el vídeo [812], continuación de éste.

[812: mxs2] Sensibilidad mixta / mixed sensitivity (2): planta generalizada y problema H-infinito asociado

Materiales: [mxs.pdf]

**** 10:58

[YouTube ▶]

Este vídeo describe las ideas básicas del método de sensibilidad mixta (*mixed sensitivity* en inglés), en controladores de un grado de libertad, y cómo transformarlo a un problema de control óptimo \mathcal{H}_∞ .

Los dos primeros minutos del vídeo revisan brevemente las fórmulas de pequeña ganancia ante incertidumbre aditiva/multiplicativa y el bucle de control de 1 grado de libertad (un resumen del vídeo [811], cuya visualización previa a éste se recomienda).

Como, en resumen, el compromiso especificaciones-robustez requiere ponderar adecuadamente el tamaño (en la frecuencia) de S (especificaciones) versus KS y T (robustez), deben introducirse unos pesos para cada una de esas funciones (o matrices, en MIMO) de transferencia. Para entender el significado en la frecuencia de esas funciones de transferencia se recomienda visualizar el vídeo [367].

Una vez planteados los pesos y su significado, se describe la planta generalizada en el problema de sensibilidad mixta, y la planta generalizada ponderada. De hecho, el problema de sensibilidad mixta es la generalización a “frecuencia” del problema estático abordado en el vídeo [449].

Con los pesos de error, salida, control se plantea un problema que con el adecuado escalado, da significado a que la norma infinito de bucle cerrado sea menor de 1.

[813: mxsd1] Sensibilidad mixta (discusión): relación tiempo/frecuencia, direccionalidad MIMO

Materiales: [mxs.pdf]

*** 07:22

[YouTube ▶]

Este vídeo discute cómo incorporar al diseño mixed sensitivity (resumido en los dos primeros minutos) especificaciones en tiempo, traducidas a especificaciones en frecuencia en el filtro W_1 . La conclusión realmente es que no se puede hacer con precisión, que hay mucha aproximación y prueba-error: el tiempo de subida está relacionado con el ancho de banda de la sensibilidad (más ancho de banda implica menos tiempo de subida, grosso modo); la sobreoscilación está relacionada con los picos de resonancia en sensibilidad S y sensibilidad complementaria T . Estas ideas son, en cierto modo, un recordatorio resumen de la interpretación de la respuesta en frecuencia de bucles cerrados de control del vídeo [367].

La última parte del vídeo argumenta que en un caso multivariable, podría haber direccionalidad tanto en las especificaciones alcanzables (de hecho, eso ya se explota en el desacoplamiento SVD, vídeo [489] o en control \mathcal{H}_∞ sin considerar robustez, vídeo [655]), como en la direccionalidad de la incertidumbre; ejemplos Matlab de esto se abordan en el vídeo [820].

[814: mxsd2] Sensibilidad mixta (discusión): formas de establecer cotas de error W_2 , W_3

Materiales: [mxs.pdf]

**** 08:17

[YouTube ▶]

Este vídeo discute aspectos acerca del modelado de la incertidumbre aditiva/multiplicativa en diseños mixed sensitivity. En concreto, comenta las ventajas de unos pesos W_2 , W_3 constantes (no incrementan el orden del regulador) y su posible interpretación. También discuten que la incertidumbre “a la entrada” (en actuadores, por ejemplo), debería multiplicarse por el número de condición y que se podría utilizar el comando `ucover` para obtener cotas con filtros no constantes de la incertidumbre (ver vídeos [803] y [819]).

[815: mxsd3] Sensibilidad mixta (discusión): problemas por cancelación y ausencia de prestaciones robustas

Materiales: [mxs.pdf]

***** 14:24

[YouTube ▶]

Este vídeo, continuación de las discusiones en los vídeos [813] y [814], presenta problemas “conceptuales” de la técnica mixed sensitivity:

- Al no haber perturbaciones en la entrada en la planta generalizada asociada, el optimizador no garantiza que las prestaciones ante ese tipo de perturbaciones sean satisfactorias. Esto está relacionado con determinadas cancelaciones controlador/planta que puede realizar `hinfsyn`. Más detalles y ejemplos sobre todo esto aparecen en el vídeo [821], y una posible solución en el vídeo [822]. Otro ejemplo sobre un control IMC se discute en el vídeo [287].
- La metodología garantiza estabilidad robusta, pero no prestaciones robustas (que para todo Δ dentro de los errores contemplados, se cumplan las prestaciones). Sólo en el caso monovariable (SISO) ante incertidumbre multiplicativa puede probarse que exigiendo un poco más al optimizador en efecto dichas prestaciones son garantizadas. El análisis en detalle de prestaciones robustas y metodologías asociadas se hace en el vídeo [835].

27.1.1 Ejemplos de código

[816: mxsml] Sensibilidad Mixta (ejemplo Matlab 2o orden)

Materiales: [[CÓD.:](#) mixsynparavideo.m] [[PDF](#)]

*** / **** 10:17

[YouTube ▶]

En este video se desarrolla un ejemplo de diseño *mixed sensitivity* en Matlab o GNU octave, para el proceso $G(s) = 2.5/(s^2 + 0.8 \cdot s + 1)$.

Se proponen unos pesos de error y de incertidumbre aditiva.

Se ejecuta el comando `mixsyn` y se presenta la respuesta en tiempo y frecuencia del diseño resultante.

Se comprueba el efecto de variar el ancho de banda (rapidez deseada) del filtro del error.

Se comprueba la estabilidad robusta ante una familia de procesos diferentes a $G(s)$ pero con error aditivo por debajo de la cota propuesta.

Se comprueba que para algunos de esos procesos, el sistema en bucle cerrado es estable (probado por pequeña ganancia) pero las prestaciones son muy malas, casi al borde de la inestabilidad (lo que demuestra con un ejemplo que la metodología de sensibilidad mixta NO garantiza *prestaciones* robustas).

[817: c3pidmx] Caso de estudio: comparativa mixedsensitivity con PIDs manuales para masa-muelle-amortiguador (Matlab)

Materiales: [[CÓD.:](#) cerml1gdmargmxsnf.mlx] [[PDF](#)]

***** 13:01

[YouTube ▶]

En este vídeo se continúa con el caso de estudio que comenzó en el análisis del compromiso prestaciones-robustez por “simulación exhaustiva” del vídeo [784] de varios reguladores PID diseñados “a mano”; la capacidad de los márgenes clásicos (fase, ganancia y retardo) para evaluar la robustez fue analizada en el vídeo [797] y, finalmente, los márgenes basados en teorema de pequeña ganancia (incertidumbre aditiva, multiplicativa, coprima normalizada) se discutieron en el vídeo [810].

Aquí se va a plantear el **diseño** por “mixed sensitivity” de reguladores \mathcal{H}_∞ que intenten alcanzar prestaciones comparables a los PIDs antes estudiados, proponiendo unos pesos para el comando `mixsyn`, analizando la estructura del regulador resultante y comparando los márgenes de robustez. De hecho, el controlador \mathcal{H}_∞ resultante puede ser expresado como un PID (minuto 8), haciendo operaciones algebraicas similares a las discutidas en el vídeo [516].

*Otra metodología de control robusto Glover-McFarlane, discutida en los vídeos [824] y [825], completará posteriormente el caso de estudio en el vídeo [827], explicando la parte final del código en el `.mlx` de materiales asociados.

Caso de estudio 2x2

[818: mxsmv]

Diseño mixed sensitivity (+ucover) en un proceso 2x2: ejemplo

Matlab

**** 10:57

Materiales:

[Cód.: mxsA2.mlx]
[PDF]

[YouTube ►]

En este vídeo se diseña un controlador mixed sensitivity (teoría en vídeo [812]) para un proceso incierto de 2 entradas y 2 salidas; un ejemplo monovariable se discute en el vídeo [816].

El proceso en concreto es el mismo que el del vídeo [801], siendo la incertidumbre estimada mediante la generación de ganancias y constantes de tiempo al azar (así como cierta dinámica de alta frecuencia).

Aquí se utiliza el comando `ucover` para obtener una cota (en función de la frecuencia) del tamaño de la incertidumbre aditiva. Otro ejemplo del uso de `ucover` en un caso monovariable aparece en el vídeo [843]. Otras opciones de uso de `ucover` en este mismo proceso se abordan en el vídeo [819].

Se comprueba cómo el ajuste al máximo error en la frecuencia depende del orden del filtro que produce `ucover`. También se comprueba que, pese a que un filtro de orden 3 tiene una cota muy ajustada, la respuesta temporal con la incertidumbre resultante es mucho más incierta que la colección de plantas al azar de la cual se partió, porque el error de modelado es “circular” y no tiene en cuenta la “direccionalidad” de la incertidumbre estructurada (paramétrica) que es mas grande en ciertas direcciones que en otras.

Tras analizar el comportamiento de `ucover`, la cota dada por este comando es directamente introducida en `mixsyn`, junto a un peso del error (inversa de una “plantilla” de la cual se desea maximizar el ancho de banda). Prueba y error iterativamente consiguen encontrar $\gamma = 1$, dando el controlador más rápido (sobre el proceso nominal) que tolera (estabilidad robusta) el error de modelado prefijado.

Por último, se reduce el orden del regulador y se comprueba la respuesta temporal y en frecuencia sobre la colección de plantas que modelaba la incertidumbre sobre la que se deseaba hacer control robusto. Se comprueba, como era de esperar, que *no* se han conseguido especificaciones robustas (que las plantas en bucle cerrado estén por debajo de la plantilla de error que el modelo nominal sí cumplía), pero que todos los bucles cerrados son estables.

[819: ucoversvd]

Ucover: ejemplo cobertura escalar/diagonal/SVD 2x2

**** 11:17

Materiales:

[Cód.: mxs2DesacSVD.mlx]
[PDF]

[YouTube ►]

En este vídeo se realizan, para un posterior diseño por sensibilidad mixta (abordado en el vídeo [820]) tres acotaciones distintas del error de modelado de un modelo nominal y una familia de plantas. En concreto, calcularemos fltros $w(s)$ o $W(s)$ para que exista Δ con $\|\Delta\|_\infty \leq 1$ de modo que:

1. Acotación escalar: $G_{real} = G + \Delta \cdot w(s)$
2. Acotación diagonal: $G_{real} = G + \Delta \cdot W(s)$, con $W(s)$ diagonal
3. Acotación SVD: siendo $\text{dcgain}(G)=USV'$, acotar $U^T G_{real} V = U^T G V + \Delta \cdot W(s)$, con $W(s)$ diagonal, de modo que a bajas frecuencias $W(s)$ acotaría la incertidumbre en las direcciones principales.

El comando usado es `ucover`. Obviamente, las opciones segunda y tercera con “direccionalidad” en la acotación del error requieren de las modificaciones conceptuales (sencillas) del teorema de pequeña ganancia discutidas en el vídeo [802].

[820: mxsmsvd]
Mixed sensitivity+ucover (2x2): desacoplamiento SVD en prestaciones
***** 10:58

Materiales: [[CÓD.:](#) mxs2DesacSVD.mlx] [[PDF](#)]
[YouTube ▶]

En este vídeo se utilizan las tres acotaciones del error mediante `ucover` detalladas en el vídeo [819], para intentar explotar una posible reducción del conservadurismo gracias a la direccionalidad del error de modelado (ideas básicas discutidas en vídeo [802]).

Con ello, se plantea resolver un problema de mixed sensitivity (`mixsyn`) de dos formas distintas:

- Primero, intentando hacer el error pequeño por igual en todas las direcciones. Realmente, esta solución coincide con la estudiada en el vídeo [818], aunque aquí se plantea la opción de utilizar otras cotas de error de modelado alternativas.
- Posteriormente, haciendo el cambio de variable $e_{svd} = U^T \cdot e$, siendo U la matriz ortogonal con las direcciones de salida en el SVD de la matriz de ganancia estática. Relajando un poco la solución anterior en la plantilla de la segunda componente de e_{svd} se puede acelerar cinco veces el ancho de banda en la primera componente (la de alta ganancia, donde en términos relativos el error de modelado tiene un efecto menor, al menos a baja frecuencia). El enfoque es parecido al desarrollado en el vídeo [655] donde las limitaciones de la acción de control eran por “saturación” en vez de “robustez”, pero las ideas básicas son compartidas.

El vídeo compara las respuestas temporales y en frecuencia nominales de referencia a error y a acción de control; también se comparan las respuestas ante perturbación a la entrada. Por último se comprueba la robustez del sistema simulando todas las respuestas temporales de la familia de plantas al azar que se usó para ser acotadas con `ucover`. Como era de esperar, todas las respuestas son estables, aunque algunas de ellas tienen alguna oscilación quizás no demasiado adecuada (se tiene estabilidad robusta, pero no *prestaciones* robustas).

27.1.2 Problemas debido a cancelaciones polo/cero y escalado

La no consideración en la planta generalizada “mixed sensitivity” de perturbaciones en la entrada $y = G(u + \delta_u) + \delta_y$ origina algunos problemas en la respuesta en bucle cerrado ante δ_u , que afectan tanto a las “prestaciones” (cuando G tiene polos lentos o poco amortiguados) como al margen de robustez ante incertidumbre coprima normalizada. Esta subsección discute dichos problemas.

[821: mxscan]
Control óptimo (sensibilidad mixta): problemas debido a cancelación de polos lentos/poco amortiguados (Matlab)
***** 11:34

Materiales: [[CÓD.:](#) mixsynconintegradoresPart1.mlx] [[PDF](#)]
[YouTube ▶]

Este vídeo ilustra, con un ejemplo Matlab, los problemas que la cancelación de polos lentos o poco amortiguados puede originar en diseños de control. De hecho, las razones fundamentales que originan esos problemas son comunes a otras filosofías de control como, por ejemplo, el control por modelo interno (ver vídeo [400]). El vídeo [287] ahonda en analizar ese tipo de problemas y lo relaciona con conceptos de controlabilidad y observabilidad de un modelo en bucle cerrado.

En concreto, se comprueba que la metodología de sensibilidad mixta **mixsyn** consigue una muy buena respuesta de error y acción de control ante cambios de referencia pero, con un proceso $G(s) = 10/(s + 0.04)/(0.01s + 1)$ la respuesta ante una perturbación en la entrada $y = G(u + \delta_u)$ es muy mala; la razón es que **mixsyn** cancela el polo $(s + 0.04)$. Si el proceso cambia a $G(s) = 10/s/(0.01s + 1)$ el bucle cerrado es ¡marginamente *inestable*! Matlab también cancela el integrador con un cero en el regulador en $(s + 10^{-7})$ y ello produce que $G/(1 + GK)$ tenga un integrador puro inestabilizando la respuesta entre δu e y en bucle cerrado (numéricamente, la simulación llega a $1.4 \cdot 10^5$ unidades de salida ante δ_u escalón unitario). Evitar esto requiere modificaciones en el planteamiento del problema, discutidas en el vídeo [822] a continuación: de hecho, si el lector va a visualizar ya dicho vídeo, la primera parte del código del mismo es coincidente con el código discutido aquí.

En el vídeo también se comprueba que estos inconvenientes no se producen cuando se desea controlar $G(s) = 10/(s + 2)/(0.01s + 1)$ con un ancho de banda de 8 radianes por segundo: la cancelación que **mixsyn** realiza del polo en $(s + 2)$ causa componentes "lentos" pero ni tienen gran amplitud ni son realmente tan lentos (sólo cuatro veces más lentos que el transitorio del error), con lo que el diseño sí que es aceptable y no requiere las modificaciones que sí son imprescindibles en procesos con un integrador o similar (por ejemplo con $G(s) = 1/(s^2 + .1s + 4)$ también se tendría problemas similares, se deja al lector dicho análisis).

Un ejemplo sobre un sistema mecánico (doble integrador) de este tipo de problemas aparece en el vídeo [833], aunque para su comprensión posiblemente sea necesario visualizar el [831] y [832] previamente.

[822: mxscan2] Problemas por cancelación de polos lentos/poco amortiguados:
solución incorporando perturbación a la entrada en planta
generalizada

Materiales: [[CÓD.:](#) mixsynconintegradoresParts1and2.mlx] [[PDF](#)]

*** 08:18

[YouTube ►]

En problemas de control óptimo con planta generalizada, la cancelación de polos lentos o poco amortiguados de la planta (cuyos problemas se ilustraron en el vídeo [821] anterior a éste) se evita incorporando una perturbación a la entrada, de modo que el control óptimo busque minimizar el error en salida (y acción de control, con la adecuada ponderación) ante un modelo $y = G(u + \delta_u) + \delta_y$. Cuanto mayor es el peso (amplitud de entrada) de δ_u para construir la planta generalizada ponderada (comparativamente con la amplitud de δ_y), más lejos está el regulador óptimo resultante de cancelar los polos lentos de G y menos norma ∞ tiene la función de bucle cerrado entre δ_u y la salida controlada.

El código de este vídeo complementa al del vídeo [821] añadiendo dicha perturbación en la entrada a la planta y comprobando numéricamente la efectividad de dicha propuesta, obteniendo respuestas razonables tanto ante cambios de referencia (equivalentemente, de perturbación δ_y) como ante escalones en δ_u .

Si estás empezando con todo esto, una explicación en detalle de las diferentes formas de construir la planta generalizada para este problema aparece en el vídeo [641] y [642]. Con dicha planta generalizada, un ejemplo sobre un sistema mecánico (doble integrador) de este tipo de problemas y la solución con perturbaciones a la entrada aparece en el vídeo [831] (planteamiento), y el vídeo [833] (comparativa entre varias opciones de diseño).

Nota: Al final, la planta generalizada (no ponderada) resultante es la que se obtiene como consecuencia del análisis sobre la incertidumbre coprime normalizada discutida en el vídeo

[806], cuyo visionado podría ser recomendable tras este vídeo si no se ha hecho ya. Como consecuencia de esta similitud, la metodología de síntesis asociada a dicha incertidumbre, discutida en el vídeo [824], tiene claras conexiones con la propuesta hecha aquí.

[823: ncfescm] Efecto del escalado en el margen ante incertidumbre coprime
normalizada: ejemplo Matlab
Materiales: [CÓD.: ncfmarginconescaladov3.mlx] [PDF]

*** 11:29

[YouTube ▶]

Este vídeo presenta un ejemplo Matlab de las cuestiones discutidas en el vídeo [809] sobre la sensibilidad de los márgenes de robustez al escalado. Se diseña una batería de reguladores por mixed sensitivity para un proceso de segundo orden, con ancho de banda creciente. Se comprueba que el margen de robustez `ncfmargin` obtenido es muy diferente según el escalado utilizado.

27.2 Control óptimo ante incertidumbre coprime normalizada, metodología Glover-McFarlane

[824: ncfsyn] Control ante incertidumbre en representación coprime
normalizada: maximización del margen de robustez
Materiales: [OptimalCFcontrol.pdf]

**** 18:34

[YouTube ▶]

Este vídeo considera la incertidumbre en factorización normalizada (equivalentemente, en el bucle con perturbaciones de entrada y salida) discutida en los vídeos [805] y [806].

En concreto, describe la planta generalizada mediante la cual un problema \mathcal{H}_∞ podría calcular el controlador que optimiza el margen de robustez ante dicha incertidumbre.

Los comandos de Matlab relacionados, utilizados en el vídeo, son `ncfmargin`, `gapmetric` (que determina la distancia entre factorizaciones) y `ncfsyn` (que computa el controlador óptimo). La interpretación en frecuencia de la distancia entre plantas, métrica “ ν -gap”, se discute en más detalle los vídeos [807] y [808].

Se presenta un ejemplo Matlab de aplicación. Se observa que al perseguir únicamente “máxima robustez” las prestaciones ante escalón de referencia no tienen por qué ser buenas. En efecto, estos controladores podrían ser un punto de comienzo razonable para primeras pruebas de control sobre procesos inestables con modelo poco preciso, pero suelen proporcionar unas prestaciones muy poco pretenciosas (dado que, obviamente, intentan maximizar robustez). El añadir requerimientos de prestaciones (rapidez, error pequeño) requiere de filtros cuyo diseño e interpretación se discuten en el vídeo [825] a continuación.

[825: gmcfl] Metodología Glover-McFarlane de conformado de bucles (loop
shaping) para control robusto
Materiales: [GMcFloopshap.pdf]

***** 17:43

[YouTube ▶]

En este vídeo se discute la metodología de conformado de bucle (“*loop shaping*”, en la literatura en lengua inglesa) propuesta por Glover y McFarlane alrededor de 1990 para incorporar requisitos de prestaciones al control de “máxima robustez” discutido anteriormente en el vídeo [824]. Esta metodología aparece con frecuencia en los libros de control robusto.

Básicamente, proponen sintetizar un control de máxima robustez ante incertidumbre coprime normalizada para una planta “escalada” o “conformada” $\hat{L} = W_2 G W_1$, siendo W_1

un peso (escalado) de entradas (paso bajo, corte en ancho de banda de actuadores), y W_2 un peso de salidas (paso bajo, corte a frecuencias bajas donde se desee error pequeño, acción integral). Mediante estos filtros se supone que el diagrama de amplitud de bucle cerrado de $\hat{T} = \hat{L}/(1 + \hat{L})$ tendría una forma satisfactoria (cerca de 1 a baja frecuencia, aproximadamente cero a alta frecuencia), de ahí su denominación de “shaping” (conformado). Obviamente, la fase y las distintas ganancias máxima/mínima/condicionamiento en el caso multivariable requieren cerrar dicho bucle con un controlador diferente a $K = 1$. En el vídeo se discuten los detalles de diversas interpretaciones teórico-prácticas de dichos filtros W_1 y W_2 .

27.2.1 Ejemplos de código

Ejemplos monovariantes

[826: gmfm1]
20:07

Control robusto Glover-McFarlane conformado de bucle:
ejemplo Matlab

**** / ****

☰

Materiales: [[CÓD.: ncfsynloopshaping.mlx](#)] [[PDF](#)]

[PLAY ▶]

En este vídeo se presenta un ejemplo en Matlab de la metodología de diseño de controladores por conformado de bucle (loop shaping) discutida en el vídeo [825].

Por simplicidad, se ha escogido un proceso de segundo orden de una entrada y una salida; un ejemplo multivariable puede verse en los vídeos [828] y [829]. Para este proceso, se seleccionan distintas opciones de pesos dependiendo de un parámetro de ancho de banda y se compara la respuesta resultante en un control de un grado de libertad, $u = K(s)y$.

En la parte final del vídeo, dado que la respuesta ante perturbación en salida (que, técnicamente, es idéntica a la de referencia) tiene mucha sobreoscilación (en la más rápida de las opciones de ancho de banda anteriores), se modifica el cálculo para que el comando **ncfsyn** produzca un regulador de dos grados de libertad, esto es, $u = K_y(s)y + K_r(s)r$. El resultado tiene una respuesta ante cambios de referencia mucho menos oscilatoria que el diseño previo de un grado de libertad (aunque, obviamente, no cambia la respuesta ante ruidos de proceso no medibles tipo escalón a la salida de la planta, si los hubiera).

***** ☰

[827: c3pidncf]

Caso de estudio: comparativa ncfsyn con mixsyn y PIDs
manuales para masa-muelle-amortiguador (Matlab)

👤 10:54

Materiales: [[CÓD.: cerml1gdmargmxsnf.mlx](#)] [[PDF](#)]

[YouTube ▶]

En este vídeo se continúa con el caso de estudio introducido en el vídeo [784] y continuado en los vídeos [797] y [810] (sólo análisis de robustez). El diseño “mixed sensitivity” fue discutido en el vídeo [817] y, ahora, se intentan conseguir prestaciones similares a los PIDs bajo discusión y al diseño mixed sensitivity mediante la metodología Glover-McFarlane, con el comando **ncfsyn**.

Ejemplo multivariable 2x2

[828: gmf2nw]

Diseño Glover-McFarlane (ncfsyn) en planta 2x2 (Matlab):
pesos unidad

👤 07:59

Materiales: [[CÓD.: ncfsA2nw.mlx](#)] [[PDF](#)]

[YouTube ▶]

Este vídeo discute el diseño Glover-McFarlane (teoría en vídeo [825]) para una planta 2x2 con polos reales.

La planta utilizada es la misma que en el vídeo [818] y, de hecho, ambos diseños son comparados (resp. temporal) en el vídeo [830].

En este primer vídeo se analiza el código para llevarlo a cabo, la acotación el error con el comando `gapmetric` y el significado de los márgenes (ver vídeos [807] y [808] para detalles sobre la interpretación de la métrica). Por simplicidad, se escogen los pesos igual a 1 (`ncfsyn` sin ponderaciones). El efecto de modificar los pesos se aborda en el vídeo [829], continuación de éste.

[829: gmf2w2] Diseño Glover-McFarlane (ncfsyn) en planta 2x2 (Matlab):
influencia de los pesos
Materiales: [CÓD.: ncfsA2.mlx] [PDF]

**** 10:52

[YouTube ▶]

Este vídeo discute el diseño Glover-McFarlane (teoría en vídeo [825]) para una planta 2x2 con polos reales. Es continuación del vídeo [828] que planteaba las ideas principales del código y el significado de los márgenes, pero con pesos unidad.

En este caso, se plantean dos filtros W_1 y W_2 paso-bajo de modo que se aumenta la magnitud de la respuesta en frecuencia a bajas frecuencias (W_2) y se disminuye a alta frecuencia (W_1 , filtro de entradas).

El efecto de W_2 es discutido, modificándolo para dar más “importancia” a la salida 1 (mejores prestaciones objetivo). También se comprueba que si se requieren unas prestaciones excesivas, entonces no se consiguen los márgenes de estabilidad mínimos para estabilizar a una familia de plantas.

[830: gmf2vsmx] Diseño Glover-McFarlane (ncfsyn) en planta 2x2 (Matlab):
interpretación loop-shaping y comparación con mixsyn
Materiales: [CÓD.: ncfsA2loopshap.mlx] [PDF]

**** 11:00

[YouTube ▶]

Este vídeo, sobre el ejemplo desarrollado en los vídeos [828] y [829], revisa la interpretación “*loop shaping*” de la metodología `ncfsyn`. Contienen un repaso a la idea teórica básica, que ya se utiliza en el diseño de compensadores (controladores) en frecuencia desde los años 1950 (en el ámbito monovariable).

Se comprueba numéricamente que el bucle abierto final $L = GK$ tiene un diagrama de valores singulares entre $G_{shaped} * \gamma$ y $G_{shaped} * 1/\gamma$, siendo γ la norma infinito mínima del comando `ncfsyn` ... aunque esa fórmula es sólo aproximada si el condicionamiento de los pesos no es cercano a uno (se comprueba numéricamente un ejemplo).

En la segunda parte del vídeo (minuto 06:50) se analiza la respuesta temporal comparada con un diseño por sensibilidad mixta (ver detalles en vídeo [818]). En ambos diseños se intenta optimizar las prestaciones hasta que permita el requisito de estabilidad robusta ante la misma familia de plantas.

Se observa que la respuesta ante referencia obtenida es más “suave” en el mixed sensitivity, pero que el regulador `ncfsyn` ha resultado más enérgico y con mayor atenuación del efecto de perturbaciones a la entrada del proceso a controlar.

27.3 Caso de estudio: doble integrador

[831: dинthi1] Control H-infinito de un doble integrador: planteamiento, planta generalizada, pesos.
Materiales: [[CÓD.:](#) dobleInthinf.mlx] [[PDF](#)]

*** 19:51
[\[YouTube ▶\]](#)

Este vídeo plantea el control de un doble integrador $1/s^2$, motivado como modelo de muchas aplicaciones de robótica, mecatrónica, vehículos, etc.

Se plantea la presencia de perturbaciones en salida (equivalentes a referencias) y en entrada (viento, fricción u otras fuerzas no modeladas), y cómo construir la planta generalizada asociada. Si no estás familiarizado con el concepto de planta generalizada y las alternativas para construirla en Matlab, podría ser de interés que visualizaras los vídeos [641] y [642].

Por último, se discuten los pesos de entradas y salidas generalizadas que dan lugar a un planteamiento final apto para el comando `hinfsyn`.

El análisis de los resultados de dicha síntesis \mathcal{H}_∞ se aborda en detalle en los vídeos [832] y [833], continuación de éste. La síntesis con estructura prefijada tipo PID se aborda en el vídeo [834].

[832: dинthi2] Control H-infinito de un doble integrador: análisis de prestaciones/robustez de regulador resultante
Materiales: [[CÓD.:](#) dobleInthinf.mlx] [[PDF](#)]

*** 20:30
[\[YouTube ▶\]](#)

Este vídeo, continuación del vídeo [831], analiza en detalle la respuesta del controlador `hinfsyn` para un doble integrador optimizado con arreglo a ciertos criterios de partida (que podrían ser diferentes, como se comenta abajo y se refiere a otro vídeo para más detalle).

Se analizan respuestas temporal, en frecuencia, márgenes de robustez clásicos (ganancia, fase, retardo), márgenes de robustez de pequeña ganancia (incert. aditiva, multiplicativa, coprima) y se hace una simulación ante error de modelado.

El vídeo [833] compara el diseño aquí detallado con otras tres opciones de pesos como alternativas de diseño.

Aparte, por su importancia tecnológica, el vídeo [834] intenta el diseño aquí detallado pero con reguladores de estructura prefijada tipo PID. Obviamente, fijar la estructura hará que no se pueda alcanzar una norma de bucle cerrado ponderada por debajo de 1.

[833: dинthi3] Control H-infinito de un doble integrador: análisis comparativo de cuatro opciones de diseño
Materiales: [[CÓD.:](#) dobleInthinf.mlx] [[PDF](#)]

*** 17:45
[\[YouTube ▶\]](#)

Este vídeo es continuación del vídeo [831] y el [832]. El último de los dos analiza en detalle la respuesta del controlador `hinfsyn` para un doble integrador (prestaciones nominales, márgenes de robustez, simulación) en lo que será el “caso 2” en el presente vídeo.

Aquí se compara el diseño allí detallado con otras opciones de pesos como alternativas de diseño.

En el diseño “ejemplo 1” sin considerar las perturbaciones a la entrada se observan los problemas de inestabilidad interna debidos a cancelaciones cerca del límite de estabilidad discutidos en los vídeos [821] y [822].

[834: dинthi4]

Control PID de un doble integrador diseñado mediante H-infinito, ejemplo Matlab: freqsep, hinfstruct

Materiales: [[Cód.:](#) dobleInthinfstruct.mlx] [[PDF](#)]

**** 14:51

[[YouTube](#) ►]

Este material, continuación del vídeo [832], intenta conseguir los objetivos del referido vídeo pero usando reguladores tipo PID (con filtro de ruido).

Se abordan dos posibilidades:

- a) Reducir el controlador `hinfsyn` a orden 2 con `freqsep` quedándose con los dos polos más lentos; en efecto, dos polos y dos ceros pueden reescribirse como un PID con filtro de ruido (vídeo [516]).
- b) Intentar minimizar la norma infinito de bucle cerrado con un regulador de estructura prefijada `tunablePID`, mediante `hinfstruct`. Otro ejemplo de uso de este comando aparece en el vídeo [845].

El primer caso obtiene un rechazo de perturbaciones excelente, tanto de entrada como de salida, y unos márgenes de fase, ganancia y retardo también muy buenos, pero a costa de una actividad de alta frecuencia excesivamente grande (dados los pesos que la limitaban), reduciendo el margen de robustez `ncfmargin` y saturando en exceso ante escalón. La norma ponderada da aproximadamente 25.

En el segundo caso, para limitar la actividad de alta frecuencia, prohibida por las plantillas elegidas, `hinfstruct` converge a un regulador de prestaciones de rechazo de perturbaciones (ancho de banda) bastante peor que los otros, debido a que el filtro de ruido lo tiene que insertar a frecuencia muy baja. La norma ponderada da aproximadamente 3.

En aplicaciones, se deberían reformular los pesos de salidas generalizadas para “moldear” las sensibilidades hasta obtener norma menor a 1; dependiendo de la aplicación, podrían estar indicadas acciones diferentes:

- 1) en posicionamiento de sistemas mecánicos con mucha fricción y “pegado” (stiction) posiblemente la actividad de alta frecuencia no sea tan perjudicial (ayuda a “despegar” las cosas para que empiecen a moverse, y el rozamiento disipa mucha energía por lo que la inestabilidad del doble integrador es más difícil).
- 2) En otras aplicaciones donde haya un control/dinámica interna (de control de ángulo/estabilización de UAVs, linealización por realimentación en robótica, dinámica de sensores o de servos) o si el sensor de posición es muy ruidoso (la realimentación de ruido de alta frecuencia puede llevar a fallos tempranos por fatiga en elementos de un mecanismo), entonces sí podría ser aconsejable limitar la actividad de alta frecuencia, a costa de reducir ancho de banda (prestaciones) o de aumentar complejidad del regulador.

El algoritmo subyacente en `hinfstruct` está inspirado en la optimización por “fuerza bruta” de reguladores (vídeos [585], [586], ...), pero con un índice de coste adaptado al problema \mathcal{H}_∞ en discusión. La ventaja es poder incorporar reguladores de estructura prefijada (PID en la mayor parte de casos prácticos), pero el inconveniente es que el tiempo de cómputo es mayor y el optimizador no siempre encuentra la solución óptima, cosa que no ocurre con el regulador de orden completo de `hinfsyn` (el cual es numéricamente mucho más rápido y preciso, y siempre converge al óptimo global en problemas bien planteados; bueno, con una cierta tolerancia).

Capítulo 28

Prestaciones robustas e incertidumbre estructurada

El Capítulo 27 discute cómo diseñar controladores que garantizan *prestaciones nominales* y *estabilidad robusta* ante incertidumbre no estructurada (aditiva, multiplicativa, coprime). Este capítulo plantea un problema de control más exigente: que se garantice no sólo estabilidad sino unas *prestaciones* mínimas satisfactorias ante errores de modelado. Finalmente, se plantea el problema con varias fuentes de incertidumbre (lo que se denomina incertidumbre estructurada). Aunque el paradigma es muy elegante y potente, la resolución numérica requiere de herramientas iterativas no convexas, que en ocasiones se quedan atascadas en soluciones muy lejanas a la realmente óptima; estos posibles problemas computacionales no estaban presentes con el control óptimo \mathcal{H}_∞ en el Capítulo 27.

28.1 Prestaciones robustas, pequeña ganancia escalado

[835: pr] Prestaciones robustas (teoría)

Materiales: [RobustPerf.pdf]

**** / ***** 19:59



- Planteamiento del problema.
- Planta generalizada 3×3 con incertidumbre, conexión LFT. Escalado –Este escalado inicial no tiene nada que ver con el teorema de “pequeña ganancia escalado” que se usará luego, ver nota al pie–.
 - Se supone que la planta generalizada está adecuadamente *ponderada* con escalados (constantes) o/y pesos en frecuencia necesarios para formar la planta generalizada 3×3 de modo que el tamaño máximo de la incertidumbre Δ sea 1, y que las prestaciones estén garantizadas si determinadas salidas tienen norma menor a 1 cuando las entradas exógenas (perturbaciones/referencias) tienen norma menor a 1.

Casos particulares:

- $\Delta \equiv 0$, es el problema del control óptimo \mathcal{H}_∞ sobre el modelo en planta generalizada “nominal”.
- Cerrando el bucle con K conocido, es el análisis de estabilidad por pequeña ganancia.
- Prestaciones robustas: planteamiento formal, $\|lft(P(\Delta), K)\|_\infty \leq 1$.
- Teorema principal: si $\|P\|_\infty \leq 1$, entonces $\|lft(P, Q)\|_\infty \leq 1$ para todo Q tal que $\|Q\|_\infty \leq 1$. NOTA: el vídeo [836] (en inglés) detalla en más profundidad esta idea.
- Corolario: diseño con prestaciones robustas mediante la optimización \mathcal{H}_∞ .
- El caso anterior es conservativo. Para mejorarlo se usa el teorema de pequeña ganancia *escalado* que modifica la planta generalizada aprovechando la existencia de multiplicadores S que verifiquen $S\Delta = \Delta S$, esto es, $\Delta = S^{-1}\Delta S = S^{-1}\Delta S$, el multiplicador no es ningún

“escalado/cambio de unidades” físico, sino una variable de decisión intermedia que reduce el conservadurismo.

En efecto, este “escalado” del teorema es la terminología tradicional en la literatura “teórica” para denominar a este importante resultado, pero no tiene el significado “físico” de los escalados/pesos en frecuencia para formar la planta generalizada que hemos mencionado arriba en el planteamiento del problema... quizás hubiera convenido en “ingeniería” de control denominarlo “teorema de pequeña ganancia con multiplicadores”.

– Enunciado del teorema, planteamiento de la optimización conjunta del controlador K y el multiplicador $S = \sigma I$, con $\sigma \in \mathbb{R}$ (el único multiplicador admisible ante incertidumbre no estructurada).

Ejemplos de aplicación directa de las ideas aquí discutidas aparecen en el vídeo [856], en los [837] y [838], así como en el caso de estudio de un depósito de los vídeos [872] y [873].

También la Robust Control Toolbox de Matlab tiene código que realiza este tipo de búsqueda de multiplicadores, con el comando `musyn`; ejemplos de uso de este comando también están incorporados en esta colección.

English version (of the central theoretical content) at video [836].

Este vídeo detalla el contenido teórico más importante del vídeo anterior (en inglés):

[836: prth1EN]
Robust performance: small-gain sufficient condition (h-infinity norm bound)

10:59

Materiales:
[RobustPerfEnglish.pdf]

[YouTube ▶]

This video presents the main “small gain” condition to ensure robust performance, the fact that the infinity norm of a given generalised plant P is less than one is a sufficient condition for robust performance (under some scalings and assumptions).

The video proves the above assertion and also relates it with the small gain conditions for robust stability and nominal performance (arising, in both cases, from a “fragment” of P).

Erratum: in audio, I often say “less than” ($<$) but slides write “lower or equal than” (\leq); the slides are correct, I just kind of abbreviate the speech but just pay attention to the correctly written expressions in the slides. Likewise, I say “negative” instead of “non-positive”; excuse my imprecise speech.

This condition will be further relaxed in forthcoming videos addressing the “scaled small gain” modification, and it will give rise to H-infinity control synthesis methods for robust performance (with so-called “multipliers” or “scalings”) conforming the basis of the mu-synthesis techniques.

A robust performance Matlab case study appears in videos [??], [??], [??].

28.1.1 Ejemplos de código

[837: prhrp]
Prestaciones robustas: rutina tma peq. Ganancia escalado

07:34

Materiales:
[CÓD.: hinfrobper.m] [PDF]

[YouTube ▶]

Este video describe cómo generar un ejemplo de código Matlab/Octave para implementar el teorema de pequeña ganancia escalado.

Revisa brevemente la teoría del video [835], y luego programa el caso de prestaciones robustas para incertidumbre no estructurada (multiplicador σI , $\sigma \in \mathbb{R}$).

Nota: El multiplicador en este código se busca por “fuerza bruta” en un rango prefijado, por simplicidad, dado que este vídeo pretende únicamente desarrollar contenido introductorio a esta temática. Existen otras opciones más eficientes en literatura (DK iteration, Matlab Robust Control toolbox, ...), por lo que este código debe ser tomado únicamente como un ejemplo “conceptual” para ilustrar la idea básica.

[838: prhrp2] Prestaciones robustas: ejemplo octave 2o orden

 23:24

Materiales: [[CÓD.:](#) prbarobper3.m] [[PDF](#)]

[[YouTube](#) ►]

En este video se resuelve en GNU/Octave un problema de prestaciones robustas ante incertidumbre aditiva no estructurada, esto es, se desea que el máximo valor singular del error esté por debajo de una determinada cota en frecuencia para todo valor de incertidumbre aditiva por debajo de otra cota en frecuencia (vídeo [835]).

En concreto, el proceso es $G(s) = 2.5/(s^2 + s + 1)$, y tiene una incertidumbre $G_{real} = G(s) + W_\delta(s)\Delta$, $\|\Delta\|_\infty \leq 1$, con $W_\delta(s) = 0.2/(s/5 + 1)/(s + 1)$.

Se utiliza como parte principal el código descrito en el vídeo [837].

Se obtiene un regulador que garantiza prestaciones robustas, y se muestran las respuestas temporales ante escalón y en frecuencia de las funciones de transferencia más relevantes del problema. Se usa tanto la planta nominal como cuatro plantas “perturbadas” dentro del margen de incertidumbre establecido, para comprobar el cumplimiento de la teoría.

Un caso de estudio de prestaciones robustas detallado se aborda en el vídeo [854], que continua en los vídeos [855] y [856].

28.2 Incertidumbre estructurada y metodologías iterativas

[839: ines1] Incertidumbre estructurada en control robusto: modelado diagonal por bloques

 09:42

Materiales: [incertestruc1.pdf]

[[YouTube](#) ►]

En este vídeo, como un primer ejemplo de incertidumbre estructurada, se modela un sistema con dos bloques de incertidumbre, uno de ellos sobre uno de los dos actuadores, y otro de incertidumbre aditiva sobre un subsistema del diagrama de bloques.

Ese ejemplo ilustra la metodología para que, en un caso general, la incertidumbre estructurada se construya en forma *diagonal por bloques*, con cada uno de los componentes de la diagonal representando una fuente de incertidumbre del proceso a controlar. Dicha incertidumbre puede ser constante, no-lineal, dinámica o/y variante en el tiempo. Para mayor detalle sobre el proceso de modelado en casos más generales, pueden consultarse los vídeos [863], [867] y, el caso de no linealidades en el vídeo [868].

[840: ines2] Incertidumbre estructurada en control robusto: teorema de pequeña ganancia escalado, técnicas iterativas

 13:18

Materiales: [incertestruc2.pdf]

[[YouTube](#) ►]

En este vídeo se discute la aplicación del teorema de pequeña ganancia escalado a la incertidumbre estructurada diagonal por bloques introducida en el vídeo [839]. De hecho, el importante problema de “prestaciones robustas ante incertidumbre no estructurada”, se convierte ahora en un caso particular de incertidumbre estructurada aquí considerada.

Se argumenta que los multiplicadores que conmutan con un Δ diagonal por bloques son, también, diagonales por bloques. Según el tipo de la incertidumbre de cada bloque, el multiplicador es constante por identidad, matriz constante completa, función de transferencia SISO por identidad o matriz de transferencia completa.

Se plantean técnicas iterativas que, alternativamente, van optimizando el controlador y el multiplicador fijado el otro. Se esboza la dificultad que plantea el uso de multiplicadores dinámicos (funciones o matrices de transferencia). Las técnicas iterativas podrían converger a diferentes soluciones dependiendo de la inicialización y de la complejidad de los multiplicadores dinámicos escogida.

28.3 Uso de la Robust Control Toolbox de Matlab, caso de estudio 1

28.3.1 Modelado

[841: rcml1]

Caso estudio Robust Control Toolbox: [1], Modelado sistemas inciertos

07:02

Materiales:

[\[Cód.: casosrobustvideo.mlx \]](#)
[\[PDF \]](#)

[\[YouTube ► \]](#)

En este vídeo se modela un sistema masa-muelle-amortiguador con constantes elásticas/fricción y dinámica de actuador inciertas, usando los comandos `ureal`, `ultidyn`. Otro ejemplo multivariable de uso de `ultidyn` aparece en el vídeo [850]. También el vídeo [891] discute el modelado de un circuito eléctrico con impedancias inciertas, cuya visualización es recomendable.

De este modo, se podrán hacer análisis y diseños de control robusto con los modelos obtenidos. El caso de estudio continua en el vídeo [843] donde se simplifica la estructura de la indertidumbre, y los vídeos [844], [845], [846], [847] donde se aplican diversas técnicas de control robusto.

[842: urealmult]

Robust Control Toolbox: multi-incidencia de parámetros inciertos, ejemplo

09:18

Materiales:

[\[Cód.: multiincidenciatst.mlx \]](#)
[\[PDF \]](#)

[\[YouTube ► \]](#)


La forma en que se escriben las ecuaciones de un modelo incierto puede causar repeticiones innecesarias de los parámetros inciertos que hagan que los cálculos sean más costosos computacionalmente y más conservadores.

Para ilustrar dicho problema, este vídeo plantea dos formas diferentes de modelar un péndulo (linealizado) en su punto de equilibrio estable con tres parámetros físicos (masa, longitud, fricción) inciertos:

1. a partir de la sustitución directa de las expresiones simbólicas de las matrices A,B,C,D del modelo linealizado en términos de elementos inciertos `ureal`
2. A partir de un diagrama de bloques donde se ha manipulado “intuitivamente” las ecuaciones intentando que los parámetros inciertos aparezcan en el menor número posible de bloques.

El resultado son dos modelos nominales idénticos, pero en el primero el total de parámetros inciertos es de *ocho* (debido a repeticiones), y en el segundo de *cuatro*. Eso hace que, por ejemplo, el comando `robstab` tarde tres veces menos tiempo en ejecutarse en el segundo caso, y que obtenga un margen de estabilidad mejor. La aplicación del comando `simplify` no consigue reducir la incertidumbre del modelo con ocho incidencias.

[843: rcml2] Caso estudio Robust Control Toolbox: [2] Simplificación de estructura de la incertidumbre

*****  09:10

Materiales: [[Cód.:](#) casosrobustvideo.mlx] [[PDF](#)]


[YouTube ▶]

Este vídeo continua el caso de estudio [841], y muestra cómo utilizar los comandos `usample` y `ucover` para simplificar la descripción de la incertidumbre a un único bloque multiplicativo o aditivo. Ello tendrá ventajas computacionales en diseño de controladores (`dksyn` o `musyn`, vídeo [846]). Nótese, sin embargo, que se cubre con una única incertidumbre un conjunto “finito” de plantas al azar, y en un conjunto “finito” de frecuencias, con lo cual podría existir una pequeña probabilidad de que alguna de las plantas “extremas” no fuera cubierta o que dicha cobertura no fuera del todo precisa a frecuencias en puntos intermedios del mallado explorado.

Otro ejemplo del uso de `ucover` en un caso multivariable aparece en el vídeo [818]. El caso de estudio continua en los vídeos [844], [845], [846], [847] donde se aplican diversas técnicas de control robusto sobre este modelo de incertidumbre.

28.3.2 Control Robusto

[844: rcml3] Caso estudio Robust Control Toolbox: [3] Diseño Sensibilidad Mixta

****  18:15

Materiales: [[Cód.:](#) casosrobustvideo.mlx] [[PDF](#)]


[YouTube ▶]

En este vídeo del caso de estudio que comenzó en el vídeo [841], se aborda el diseño *mixed sensitivity*, primero en su versión “original” (prestaciones nominales + estabilidad robusta, $\gamma = 1$), que se comprueban con el comando `robstab`, y luego relajando los requerimientos para conseguir prestaciones robustas (`wcgain` menor a uno de la planta generalizada ponderada).

Los comandos `robstab`, `wcgain` y `robgain` son discutidos en profundidad en el vídeo [852] en el contexto de control robusto, aunque una primera aproximación “intuitiva” se hace en el vídeo [851] sobre un masa-muelle-amortiguador. Si los rangos de variación alrededor del valor nominal de los parámetros inciertos son asimétricos, los márgenes `robstab` deben ser interpretados adecuadamente, ver vídeo [853].

El diseño de PID's para mixed sensitivity se aborda en el vídeo [845], y el diseño mu-síntesis en los vídeos [846] y [847], continuación del caso de estudio.

[845: rcml4] Caso estudio Robust Control Toolbox: [4] Diseño PID's con hinfstruct

****  10:29

Materiales: [[Cód.:](#) casosrobustvideo.mlx] [[PDF](#)]

[YouTube ▶]

Este vídeo aborda el mismo problema que el vídeo [844], pero con un regulador PID. La estructura se fija con el comando `tunablePID`, y se optimiza con `hinfstruct`. Otro ejemplo de esta metodología de sintonizado aparece en el vídeo [834] sobre un modelo doble integrador $1/s^2$.

El caso de estudio continua en el vídeo [846] con mu-síntesis. El diseño de PID's con mu-síntesis también es posible (y de hecho puede que sea mejor al menos teóricamente que lo que se hace aquí), eso se aborda en el vídeo [847].

El algoritmo subyacente está inspirado en la optimización por “fuerza bruta” de reguladores (vídeos [585], [586], ...), pero con un índice de coste específicamente adaptado al problema \mathcal{H}_∞ en discusión aquí. La ventaja es poder incorporar reguladores de estructura prefijada (PID en la mayor parte de casos prácticos), pero el inconveniente es que el tiempo de cómputo es mayor y el optimizador no siempre encuentra la solución

óptima, cosa que no ocurre con el regulador de orden completo de `hinfsv` (el cual es numéricamente mucho más rápido y preciso). De hecho, otra opción para conseguir un regulador final en forma PID sería diseñar un regulador \mathcal{H}_∞ con `hinfsv`, y posteriormente tratar de aproximarlos a un regulador de orden 2 para escribirlo como un PID (vídeo [516]).

Nota: la optimización “genérica” numérica también puede aplicarse a minimizar el máximo coste de una serie de simulaciones con una familia de plantas: lo que en el vídeo [784] era un simple análisis puede convertirse en un problema de optimización *minimax* para ser abordado con las adecuadas herramientas. Se recomienda al lector revisar las ideas del vídeo [789] (y los resultados del vídeo [791]) como enfoque alternativo (o complementario) basado en especificaciones *temporales*, en vez del `hinfstruct` basado en especificaciones puramente *frecuenciales*. Con las adecuadas modificaciones, podría optimizarse un índice o plantear restricciones combinando ambos aspectos temporal-frecuencial (en una planta sin incertidumbre, la incorporación de restricciones sobre la respuesta en frecuencia en Simulink se aborda, por ejemplo, en el vídeo [588]). Se propone al lector interesado que desarrolle por su cuenta estos aspectos.

[846: rcml5]

Caso estudio Robust Control Toolbox: [5] Diseño iterativo usando mu-síntesis (`musyn`, `dksyn`)

Materiales: [[Cód.:](#) casosrobustvideo.mlx] [[PDF](#)]

12:42

[YouTube ▶]

Este penúltimo vídeo del caso de estudio aborda la optimización mediante la iteración entre multiplicadores y controladores que sugiere el teorema de pequeña ganancia escalado. El comando Matlab que lo realiza es `dksyn`. El uso de la descripción reducida de la incertidumbre generada con `ucover` en el vídeo [843] conlleva ventajas computacionales, a cambio de ser “conservativo” para algunas plantas (cubrir con una “bola” una geometría “estructurada” diferente) y, a la vez, “arriesgado” debido a haber seleccionado un conjunto finito de plantas en un conjunto finito de frecuencias para la acotación con `ucover`.

El diseño de reguladores PID con mu-síntesis se aborda en el vídeo [847], que concluye el caso de estudio.

Nota: a partir de la versión R2020a, el comando recomendado es `musyn` que sustituye a `dksyn` (obsoleto), con una sintaxis similar. Este vídeo se grabó con anterioridad a dichos cambios. El comando `musyn` se usa, por ejemplo, en el vídeo [850] donde se aplica a un proceso multivariable de 2 entradas y 2 salidas. El material MLX ha sido actualizado.

Otro caso de mu-síntesis se aborda en el vídeo [857]; el diseño de PID con mu-síntesis se discute en el [858].

[847: rcml6]

Caso estudio Robust Control Toolbox: [6] Diseño de PID con mu-síntesis (`musyn`)

Materiales: [[Cód.:](#) casosrobustvideoconPIDmu.mlx] [[PDF](#)]

10:59

[YouTube ▶]

Este vídeo concluye el caso de estudio que comenzó en el vídeo [841] (modelado), el [843] (simplificación de la incertidumbre), [844] (mixed-sensitivity sobre planta nominal), [845] (PID con `hinfsv`) y [846] (mu síntesis).

En este vídeo, diseñaremos un controlador PID con `musyn`, porque también lo permite, como el `hinfstruct` que se usó en el caso del vídeo [845].

El código es bastante parecido al caso `hinfstruct`, realmente, similar al vídeo [??].

La parte final del vídeo compara las prestaciones obtenidas con cada una de las metodologías.

Existen discrepancias en los resultados obtenidos con versiones anteriores de Matlab, no sé si por error mío en la copia/adaptación del código o por mejoras internas en la Robust Control Toolbox.

28.3.3 Análisis y diseño de PIDs

[848: mmimod] Modelado de un sistema incierto para control PID robusto y planteamiento del problema

Materiales: [CÓD.: PIDaojoLightlyDamped.mlx] [PDF]

*** 10:31

[YouTube ▶]

Este vídeo discute el planteamiento del problema de control robusto (con PID, ese será el objetivo del vídeo [849], continuación de éste). Se introducen amortiguamiento, rigidez y retardo (fase no mínima) inciertos, se calcula el ancho de banda conseguible (nominalmente, sin tener en cuenta limitaciones del compromiso especificaciones-robustez). También se introducen otro par de polos resonantes a alta frecuencia para comprobar que los diseños del vídeo arriba referido son robustos ante ese tipo de dinámica no modelada.

[849: mmipid] Control PID de sistema incierto: prueba de robustez de varias metodologías, y sintonía manual (Matlab)

Materiales: [CÓD.: PIDaojoLightlyDamped.mlx] [PDF]

**** 18:55

[YouTube ▶]

Este vídeo, continuación del [848], plantea distintas posibilidades de regulador PID para el sistema incierto modelado en el vídeo referido.

En concreto, se prueban prestaciones nominales y robustas de un regulador proporcional de ganancia 1, de un PD 1+s (añadiendo un filtro que corta a 9 rad/s), de un PID basado en cancelación (similar a lo que un IMC-PID hubiese dado con modelo de referencia $\frac{1}{(s/3+1)^2}$), del `pidtune`, y de una modificación “a ojo” para intentar mejorar las cosas.

El diseño final debe tener prestaciones robustas ante simulación exhaustiva de la incertidumbre, así como buenos márgenes de robustez `robstab`, `allmargin` y `ncfmargin`.

Diseños PID basado en optimización de norma infinito o márgenes de robustez (`hinfstruct`, `musyn`) se discuten en los vídeos [845] y [847].

28.3.4 Caso multivariable

[850: musyn2] Diseño μ -síntesis para una planta 2x2 y comparación con mixsyn

Materiales: [CÓD.: musyn2x2.mlx] [PDF]

**** 13:59

[YouTube ▶]

Este vídeo presenta el diseño de dos controladores para una misma planta de dos entradas y dos salidas. La planta es la misma que la considerada en el vídeo [818] y, de hecho, la cobertura de incertidumbre aditiva no estructurada es la resultante del `ucover` de dicho vídeo. No obstante, aquí *no* se considera una “familia de plantas” y se plantea directamente la construcción de un modelo incierto $P_r = G(s) + w(s)\Delta(s)$ siendo Δ un bloque `ultidyn` de norma 1, y $w(s)$ un peso escalar que modula el tamaño del error en función de la frecuencia.

Las dos estrategias de control a comparar son `mixsyn` y `musyn`, buscando maximizar el ancho de banda donde el error es pequeño mediante una plantilla generada con `makeweight`, invirtiéndola posteriormente. Un caso monovariable de μ -síntesis se aborda en el vídeo [846], usando la versión antigua del comando que se denominaba `dxsyn`.

El `mixsyn` consigue estabilidad robusta, pero no prestaciones robustas, como se comprueba con `robstab`, `wcgain` y `robgain`. Otros ejemplos del uso de esos márgenes aparecen en el vídeo [852].

Para que el `musyn` obtenga una solución que garantice prestaciones robustas, el ancho de banda debe disminuirse respecto a la solución `mixsyn`, más “fácil” porque sólo buscaba

prestaciones nominales. Haciendo nuestros requerimientos aproximadamente el doble de lentos, se consigue una solución factible: `musyn` consigue estabilidad robusta (`robstab` da 1.4), y prestaciones robustas (tanto `wcgain` como `robgain` dan un resultado de 1). Una presentación del significado de esos comandos aparece en el vídeo [??].

Una vez que se comprueba que la respuesta temporal y en frecuencia del bucle diseñado con μ -síntesis es satisfactoria, se procede a reducir el orden del controlador, a partir del minuto 11:00, con `balred`. Originalmente era de orden 26, pero se reduce a orden 5 consiguiendo un `wcgain` de 1.07 y un `robgain` de 0.96 sin modificar el margen `robstab`: las prestaciones son robustas para el 96% de las plantas en la bola de incertidumbre y en las que no lo son, la plantilla se sobrepasa como mucho en un 7%, lo que se considera satisfactorio.

28.3.5 Significado de márgenes de robustez

[851: stabmrg] Márgenes de estabilidad `robstab`, `wcgain`, `robgain`: ejemplo sistema 2o orden amortiguamiento incierto

Materiales: [CÓD.: `robstabwcgainOL.mlx`] [PDF]

**** 10:50

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute el significado de los márgenes de estabilidad robusta (o de prestaciones robustas) dadas por los comandos `robstab`, `wcgain` y `robgain` de la Robust Control Toolbox de Matlab.

Por simplicidad, se aborda el caso de un sistema en bucle abierto incierto de segundo orden (masa-muelle-amortiguador con amortiguamiento incierto), aunque realmente el uso de los comandos más habitual es para problemas de diseño de control robusto h-infinito y μ -síntesis; pero aquí el objetivo es entender el resultado con un sistema que comprendemos físicamente, para comprobar que coinciden con lo intuitivamente esperado. Un análisis del uso de estos tres comandos en ese contexto se aborda en el vídeo [852], cuya visualización a continuación de este se aconseja.

[852: margesc] Márgenes de estabilidad ante incertidumbre estructurada (Matlab): `robstab`, `wcgain`, `robgain`

Materiales: [CÓD.: `AnálisisPeqGanEscaladoMu.mlx`] [PDF]

**** 13:24

[YouTube ▶]

Este vídeo analiza, en tres diseños de control diferentes (PID, `mixsyn` y `ncfsyn`), los resultados de los comandos de la Robust Control Toolbox de Matlab para determinar márgenes de estabilidad y prestaciones robustas. Si estás empezando en estos temas, quizás te interese visionar primero el vídeo [851] donde se explican dichos comandos en un contexto más simple (masa-muelle-amortiguador incierto, pico de resonancia).

Los detalles de los diseños Glover-McFarlane y Sensibilidad Mixta se pueden consultar en los vídeos [829] y [818].

La incertidumbre sobre una familia de plantas se ha acotado con `ucover`, como aditiva no estructurada. No sería necesario y, salvo por razones computacionales, lo lógico sería acotar la familia de plantas con elementos `ureal` y `ultidyn`, como en el vídeo [841]. Por simplicidad computacional, y para comparar con las fórmulas de pequeña ganancia, se ha elegido `ucover`.

Por hacer al vídeo autocontenido, se ha dedicado hasta el minuto 04:50 del mismo a resumir los diseños de controladores y acotación de la incertidumbre pero, como ya se ha mencionado, se remite a los vídeos [829], [818] y [830] para detalles. Aquí se ven rápidamente y se calculan las cotas de incertidumbre que serían admisibles por teorema de pequeña ganancia (no escalado) ante incertidumbre aditiva.

Los comandos analizados son:

- **robstab**, que determina el porcentaje de incertidumbre tolerada hasta que una planta en un modelo incierto puede hacerlo inestable... si el resultado es mayor de 1, hay estabilidad robusta.
- **wcgain**, que determina la ganancia máxima en bucle cerrado con la incertidumbre del modelo... si no hay estabilidad robusta, es infinita... si hay prestaciones robustas, es menor de uno.
- **robgain**, que determina el porcentaje de incertidumbre tolerada para tener prestaciones robustas... si es menor de 1 es que no se garantizan prestaciones para la incertidumbre del modelo... si es cero, no se garantizan ni siquiera para la planta nominal.

Nota: si los rangos de variación de los parámetros inciertos son asimétricos tomando el valor nominal como centro, los valores numéricos de márgenes **robstab** y **robgain** deben ser interpretados adecuadamente, dado que Matlab transforma internamente todos los intervalos de incertidumbre al intervalo $[-1, 0$ (nominal), $+1]$, ver vídeo [853] para detalles.

[853: asimrc] Interpretación de márgenes de estabilidad con rangos asimétricos en Robust Control Toolbox

Materiales: [[CÓD.:](#) UncertainNonSymmetric.mlx] [[PDF](#)]

*** 07:42

[YouTube ▶]

En este vídeo se discute la interpretación de los márgenes proporcionados por **robstab** o **robgain** cuando se tienen elementos, por ejemplo **ureal**, con rangos asimétricos como `ureal('Q',5,'PlusMinus',[-1,+3])`.

En ese caso, Matlab hace una transformación a un intervalo de modo que el valor nominal corresponda a 0, y los extremos correspondan a ± 1 , ver el comando **actual2normalized**. Esa transformación tiene la forma $F(\rho) = (a + b\rho)/(c + d\rho)$ y es, por tanto, no lineal en un caso general (aunque es lineal en el caso simétrico alrededor del nominal).

En este vídeo se representa gráficamente dicha transformación y se discute su significado para entender correctamente los intervalos “físicos” de garantía de estabilidad dado un cierto margen **robstab**.

28.4 Caso de estudio 2 prestaciones robustas

[854: cerp1] Prestaciones robustas, caso estudio Matlab: 1- planteamiento del problema

Materiales: [[CÓD.:](#) RPCE1.mlx] [[PDF](#)]

*** 09:19

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

En este video se empieza a discutir un análisis de prestaciones robustas de un proceso $5/(s+1)^2$. Este vídeo, muy breve, simplemente presenta un PID diseñado por prueba y error. Se analizan sus prestaciones nominales de error de bucle y de referencia a acción de control, tanto en tiempo como en frecuencia.

Si no tienes familiaridad con los conceptos de respuesta en frecuencia en bucle cerrado, te recomendaría visualizar el vídeo [367] antes que éste.

Se consideran satisfactorias y se plantea el problema de cómo garantizar, ante una cierta estimación del error de modelado (incertidumbre aditiva no estructurada, de cota de tamaño 0.2), que la respuesta del sistema ante error de modelado se “parece” a la simulación del nominal (garantizar prestaciones robustas). La solución de dicho problema se discutirá en los vídeos [855] y [856], continuación de éste.

El vídeo [838] aborda otro caso de estudio similar a éste.

[855: cerp2]	Prestaciones robustas, caso estudio Matlab: 2- planta generalizada y análisis de un PID Materiales: [CÓD.: RPCE1.mlx] [PDF]	****  18:35 [YouTube ►]
--------------	--	---

*[LINK TO ENGLISH VERSION](#)

Este vídeo es continuación del vídeo [854], donde se analizaba un bucle de un proceso de 2o orden con un PID y se consideraba satisfactorio al menos sobre el modelo nominal, pero se planteaba la pregunta de cómo garantizar que ante un cierto error la respuesta continuaba siendo suficientemente parecida a la simulación nominal.

En este vídeo vamos a plantear cómo garantizar dichas prestaciones robustas.

Primero, se prueba la fórmula de “estabilidad robusta ante incertidumbre aditiva” que se obtiene del teorema de pequeña ganancia. Se comprueba que, en efecto, se tiene estabilidad robusta y que hay un margen de robustez “extra” (podría el error ser 2.5 veces más grande manteniendo estabilidad) que quizás nos permita garantizar algo más que la mera estabilidad en bucle cerrado (claro, las prestaciones que buscamos).

A continuación, se cuantifica qué entenderemos por prestaciones robustas: visto el bodemag del error ante cambios referencia nominal, se diseña un límite con un filtro **makeweight** un poco por encima de dicho error nominal, y se prescribe que el error de modelado debe mantener el error de bucle por debajo de ese límite. Por simplicidad, no se consideran limitaciones en acción de control (robustez ante incertidumbre aditiva ya la limita, en cierto modo).

El siguiente paso es plantear la planta generalizada con incertidumbre asociada al problema. Asimismo, el teorema que se usa para probar prestaciones robustas requiere normalizar a 1 la cota del error de modelado, y normalizar a 1 las prestaciones deseadas, por lo que se construyen unos ciertos pesos. Con ello, si la norma infinito de la planta generalizada es menor a 1, se prueban las prestaciones robustas.

Con los pesos elegidos, NO se consigue dicha norma menor a 1, pero moviendo el tamaño de la incertidumbre de los pesos de entrada a los pesos de salida resulta un problema equivalente, que SÍ tiene norma menor a 1 y prueba prestaciones robustas. Ello muestra que en el problema hay un grado de libertad adicional que puede representarse con un multiplicador S en las fórmulas de teorema de pequeña ganancia escalado.

El vídeo [856], continuación de éste, hará la simulación de estas prestaciones y comparará el PID con otro regulador **hinfsyn** que intente mejorar las prestaciones que se garanticen.

[856: cerp3]	Prestaciones robustas, caso estudio Matlab: 3 - pequeña ganancia escalado, control h-infinito, simulación Materiales: [CÓD.: RPCE1.mlx] [PDF]	*****  14:21 [YouTube ►]
--------------	--	--

*[LINK TO ENGLISH VERSION](#)

En este vídeo se concluye el caso de estudio que comenzó en el vídeo [854] (planteamiento del problema), y continuó con el vídeo [855] (construcción de planta generalizada y análisis de prestaciones de un PID sintonizado por prueba y error).

Aquí se plantea usar el teorema de pequeña ganancia escalado para obtener resultados menos conservativos.


Primero, variando el multiplicador de modo que se puede probar prestaciones robustas para un error de modelado algo más grande para el PID.

Segundo, usando el controlador **hinfsyn** que minimiza la norma de la planta generalizada asociada al problema de prestaciones robustas. Cambiando también el multiplicador de forma iterativa (manual) conseguimos garantizar prestaciones robustas ante mayor tamaño de error de modelado y con un ancho de banda algo más rápido que con el PID (aunque, bueno, el PID podría haberse sintonizado con **hinfstruct** en vez de por prueba y error para intentar mejorarlo).

En el vídeo también se presenta la simulación en tiempo y frecuencia de las distintas opciones de diseño ante plantas aleatorias dentro de las cotas de error de modelado (usando la Robust Control Toolbox).

Nótese que, al no haber perturbación a la entrada, el controlador cancela totalmente la planta (eso podría dar lugar problemas en determinadas situaciones que podrían requerir modificar la planta generalizada), ver vídeo [822]).

Nota: El objetivo de este vídeo es didáctico sobre los conceptos preliminares en el problema de prestaciones robustas: la búsqueda “manual” del multiplicador es muy tediosa y sujeta a errores y, por tanto, NO se recomienda hacer las cosas como aquí se han hecho. Como poco, la búsqueda no resulta difícil de automatizar incluyéndola en un bucle (ver caso de estudio en vídeos [837] y [838] en un problema muy parecido a éste); bueno, lo recomendable sería dejar toda la construcción de planta generalizada y búsqueda de multiplicadores a la Robust Control Toolbox en el comando **musyn** (μ -síntesis), dado que para controlador fijo, buscar el multiplicador que minimiza la norma infinito es un problema convexo. Esto se hace, continuando este caso de estudio, en los vídeos [857] para regulador de estructura arbitraria state-space, y el vídeo [858] optimizando un regulador de estructura prefijada tipo PID. Realmente la metodología del vídeo [855] y de éste deben ser consideradas sólo a título didáctico para comprender mejor qué hace **musyn**, pero la estrategia recomendable es usar **musyn** directamente, como en los vídeos continuación de este caso arriba referidos.

[857: cerp4mu] Prestaciones robustas, caso estudio Matlab: 4- μ síntesis *****  15:16
Materiales: [[CÓD.:](#) RPCE1musyn.mlx] [[PDF](#)] [YouTube ▶]

* [LINK TO ENGLISH VERSION](#)


Este video continua el caso de estudio de los vídeos [854], [855] y [856]. En estos vídeos, se planteaba el problema de prestaciones robustas, se formaba una planta generalizada con incertidumbre, y se buscaban multiplicadores (escalares) de forma manual, por prueba y error.

Aunque de interés a efectos didácticos de cara a comprender mejor el problema de prestaciones robustas y el teorema de pequeña ganancia escalado, realmente rara vez hay que hacer las cosas tal y como se dice en dichos vídeos, porque de todo ese detalle se encarga el código interno de **musyn**, μ -síntesis. En este vídeo se compara la planta generalizada que construye **musyn** al ser ejecutado (podemos hacerlo de forma explícita con el comando **lftdata**) con la planta que construimos manualmente en el vídeo [855], comprobando que coinciden.

Luego, la búsqueda de multiplicador se hace en una línea con **musyn**; el multiplicador para incertidumbre LTI puede ser dinámico porque si $\Delta(s)$ es SISO entonces $\Delta(j\omega) \cdot D(j\omega) = D(j\omega) \cdot \Delta(j\omega)$ y se puede usar en teorema de pequeña ganancia escalado. El comando **musyn** con opciones por defecto usa aquí un $D(s)$ de orden 8, y produce un regulador de orden 11.

La parte final del vídeo discute la reducción de orden de dicho regulador con **balred** y **freqsep** hasta orden 2. El resultado es muy parecido a un PID, de modo que en el vídeo [858] continuación de este, se usa **musyn** para optimizar directamente los parámetros de un PID, dando un resultado muy parecido.

Por brevedad aquí, el regulador μ -síntesis de este vídeo y el PID del citado vídeo siguiente son ambos simulados en la parte final del vídeo [858].

[858: cerp5mu] Prestaciones robustas, caso estudio Matlab: 5- μ síntesis, ajuste de PID *****  09:06
Materiales: [[CÓD.:](#) RPCE1musyn.mlx] [[PDF](#)] [YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo continua con el caso de estudio iniciado en el vídeo [854], en concreto optimizando los parámetros de un PID mediante **musyn** para comparar con el resultado de un regulador state-space genérico usado en el vídeo [857], cuya visualización previa es aconsejable (los tres primeros minutos de este vídeo resumen brevemente los vídeos anteriores).

Se comprueba que el resultado es muy parecido al del vídeo [857] y que son satisfactorias las simulaciones tanto en tiempo como en frecuencia, con plantas al azar dentro de la familia con incertidumbre aditiva para la que se ha diseñado todo. Con esto se concluye el caso de estudio.

Un ajuste similar de PIDs con mu-síntesis se aborda en el vídeo [847].

28.5 Caso de estudio 3, observador con musyn

[859: obsmu1] **Modelado LFT de sistema con amortiguamiento incierto para diseño robusto de observadores**

Materiales: [[CÓD.:](#) MuvsKalvsHinfiltter.mlx] [[PDF](#)]

**** 12:30

[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo aborda el modelado de un sistema masa-muelle-amortiguador con coeficiente de amortiguamiento incierto, al que le entra una fuerza con ancho de banda limitado (o sea, que será la salida de un filtro de primer orden). Se detalla el modelado LFT dejando la incertidumbre separada de la planta, y las salidas que serán necesarias para resolver un problema de diseño óptimo de estimadores de la fuerza. Aquí se insiste en el concepto de modelado LFT de la incertidumbre, porque lo considero un objetivo didáctico importante per se... pero en este caso de incertidumbre paramétrica real, le podemos dejar la faena a Matlab de montar la LFT y hacerlo mucho más sencillo todo, como se ve para este mismo sistema en el vídeo [851].

Otro ejemplo en la misma línea de modelado de un sistema eléctrico incierto se aborda en el vídeo [864].

El caso nominal, esto es, sin incertidumbre, y en tiempo discreto, fue objeto de estudio de los vídeos [668], [669] y [670] que, dependiendo de tu familiaridad con estos temas, podrías querer visualizar previamente, antes de pasar al diseño con incertidumbre que abordamos en el vídeo [860], en tiempo continuo.

[860: obsmu2] **Estimación fuerza sobre sistema mecánico incierto: observador mu-síntesis (Matlab) + h2 y hinf nominales**

Materiales: [[CÓD.:](#) MuvsKalvsHinfiltter.mlx] [[PDF](#)]

**** 17:30

[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo aborda el diseño de un estimador óptimo (en cierto sentido) de la fuerza que actúa sobre un sistema masa-muelle-amortiguador con coeficiente de amortiguamiento incierto.

Aunque aquí se repasa rápidamente, una explicación más extensa de la parte de modelado LFT incierto se aborda en el vídeo [859], que podría ser aconsejable visualizar dependiendo de tu familiaridad previa con estos temas.

En este vídeo, se plantean plantas generalizadas y plantas generalizadas ponderadas para abordar el problema. Realmente, el caso nominal, esto es, sin incertidumbre, fue objeto de estudio de los vídeos [668], [669] y [670] aunque en ese caso se abordó en tiempo discreto, pero los pesos en frecuencia eran bastante similares a los discutidos aquí.

Aquí revisaremos los diseños **h2syn** y **hinfsyn** sobre la planta nominal, y el diseño **musyn** sobre la planta con incertidumbre.

Se analizan las prestaciones en términos de márgenes **robstab** (no relevante en problemas de “observadores”), **wcgain** y **robgain**.

El resultado de **musyn** es un estimador de orden elevado. La reducción de orden y la simulación en tiempo y frecuencia se abordarán, por brevedad, en el vídeo [861], continuación de este.

[861: obsmu3]

Estimación fuerza sistema mecánico incierto: análisis
prestaciones h2,hinf,musyn en frecuencia

Materiales: [[CÓD.:](#) MuvsKalvsHinfILTER.mlx]

****  10:45

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo continua el caso de estudio del diseño de un estimador de una variable interna (observador) de un sistema con modelo incierto, que comenzó en el vídeo [859] (modelado), continuando en el [860] (planta generalizada y diseño óptimo h2,hinf,mu).


Formalmente la teoría persigue garantías en el dominio de la frecuencia, por lo que en ese vídeo se discute el comportamiento del observador en el dominio de la frecuencia, tanto nominalmente como ante error de modelado, comparando los resultados **h2syn**, **hinfsyn** y **musyn**.

Como lo que se verá en una aplicación real son señales en el dominio del tiempo, dedicaremos un vídeo ([862]) que concluirá el caso de estudio con simulaciones en el dominio del tiempo incorporando ruidos de medida.

[862: obsmu4]

Estimación fuerza sistema mecánico incierto: prestaciones
h2,hinf,musyn en dominio del tiempo

Materiales: [[CÓD.:](#) MuvsKalvsHinfILTER.mlx] [[PDF](#)]

*****  21:39

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo es la conclusión del caso de estudio del diseño de un estimador de una variable interna (observador) de un sistema con modelo incierto, que comenzó en el vídeo [859] (modelado), continuando en el [860] (planta generalizada y diseño óptimo h2,hinf,mu) y en el [861] (análisis en frecuencia de los resultados).


Aunque formalmente la teoría persigue garantías en el dominio de la frecuencia, lo que realmente se verá en una aplicación real son señales en el dominio del tiempo, por lo que dedicamos este vídeo a discutir dichas prestaciones ante escalón y ante entradas senoidales con ruido de medida aleatorio. Se discute que como el hinfsyn o musyn no optimizan la varianza de la salida, habría que comprobar experimentalmente o/y añadir filtrado de alta frecuencia a las plantillas límite para asegurar que la varianza ante ruido de medida es aceptable. El caso en tiempo continuo tiene además sus peculiaridades porque el ruido blanco es una señal de varianza infinita... El objetivo de este vídeo no es entrar en esas sutilezas, pero al menos acercarse a una simulación razonable de todo eso.

Capítulo 29

Modelado de sistemas inciertos/no lineales para control robusto

29.1 Modelado Linear-Fractional-Transformation

[863: lftr1] Modelado LFT para Control Robusto: 1.- Planteamiento, introducción general 1 parámetro lineal
Materiales: [ModeladoLFT.pdf]


*****  27:38
[YouTube ▶]

Este vídeo plantea el problema del modelado linear fractional transformation (LFT) de sistemas inciertos y no-lineales, para obtener un diagrama de bloques con incertidumbre para diseño de control (como, por ejemplo, el considerado en los videos [835] y [837]).

El video describe:

- El concepto matemático de transformación lineal fraccional (LFT), en sus variaciones LFT “inferior” y LFT “superior”. *Nota:* estas interconexiones son un caso particular del star-product que el comando `lft` de Matlab puede manejar, con las fórmulas que se discuten en el vídeo [285].
- El diagrama de bloques asociado a una LFT en problemas de control.
- Revisa el concepto de planta generalizada como caso particular de la interconexión LFT, como se discutió en los vídeos [639] y [640].
- Equivalencia entre representación interna y diagrama LFT con s^{-1}
- Realización de una función de transferencia $G(s)$: pasa de $G(s)$ a representación interna y, por tanto, a diagrama LFT.
- Modelos algebraico-diferenciales de índice 1. Modelos cuasi-lineales.
- Ejemplo: representación LFT de una ecuación lineal incierta $z = (\theta + \delta)w$, siendo θ el valor nominal de un parámetro y δ la “incertidumbre paramétrica”.
- Aplicación del ejemplo anterior para plantear un problema de modelado un circuito eléctrico con una resistencia incierta, que será resuelto en el video [864].

[864: lftr1ml] Modelado LFT para Control Robusto; parte 2 Ejemplo MATLAB 1 parámetro lineal circuito eléctrico
Materiales: [ModeladoLFT.pdf]

*****  23:42
[YouTube ▶]

Este video modela un circuito eléctrico con resistencias y condensadores donde una de las resistencias es “incierto”, $R_{2,real} = R_{2,nominal} + \Delta_{R_2}$, extrayendo Δ_{R_2} en un diagrama LFT.

Otro ejemplo Matlab sencillo de un sistema masa-muelle-amortiguador con amortiguamiento incierto se aborda en el vídeo [859]; sería interesante que visualizaras dicho ejemplo antes de pasar a casos más complejos.

Utiliza el Symbolic toolbox de Matlab para manipular las ecuaciones, tanto del modelo nominal como del modelo incierto.

Mediante la representación LFT se obtiene un margen de estabilidad por pequeña ganancia (incrementos de resistencia admisibles). Obviamente, el teorema de pequeña ganancia no escalado es conservativo y el margen real es mucho mayor (pequeña ganancia incluye posibles dinámicas, retardos, no-linealidades que una resistencia “constante pero incierta” no va a tener; tampoco considera el hecho que la resistencia es un elemento “pasivo” que disipa energía y contribuye a estabilizar; las modificaciones necesarias para incorporar ese conocimiento se discuten en los vídeos de esta colección sobre *pasividad*).

[865: lftr1rac] **Modelado LFT para Control Robusto: 3, Formas racionales-polinomiales genéricas de 1 parámetro**

Materiales: [ModeladoLFT.pdf]

***** 31:53

[YouTube ►]

Este video discute

- Revisión del modelado de una incertidumbre aditiva en un coeficiente, video [863].
- Modelado de una incertidumbre porcentual en un coeficiente $z = \theta(1 + \delta)w$.
- Parámetros repetidos: modelado de varias ecuaciones lineales que dependan de un único parámetro incierto $a = \delta Mb$, basado en el SVD de M , y el LFT con δI , tamaño de I igual al rango de M .
- Incertidumbre sobre la inversa: $z = (\theta + \delta)^{-1}w$.
- Generalización 1: expresiones polinomiales en parámetros inciertos $z = (\theta + \delta)^2w$, como LFT con $\delta I_{2 \times 2}$.
- Generalización: expresiones racionales en parámetros inciertos, $z = p(\delta)/q(\delta) \cdot w$. Esbozo de la metodología, y planteamiento de un problema a resolver en vídeo separado (vídeo [866]).

[866: lftr1racml] **Modelado LFT para Control Robusto: 4 Ejemplo MATLAB formas racionales genéricas**

21:19

Materiales: [[CÓD.](#): genrationallFT.mlx] [[PDF](#)]

[YouTube ►]

Este vídeo detalla cómo obtener una representación LFT del sistema incierto:


$$\frac{dx}{dt} = \begin{pmatrix} \frac{3}{1+\delta^2} & -2+\delta \\ -4+\delta^2 & \frac{\delta}{1+\delta+\delta^2} \end{pmatrix} x + \begin{pmatrix} -1 \\ 5+\delta \end{pmatrix} u, \quad y = (1 - \delta)x$$

mediante manipulaciones con la Control System toolbox de Matlab, utilizando el comando **ss** que transforma de expresión racional a variable de estado para modelado LFT (es un uso diferente del “oficialmente” pensado, ahora s será una función de δ y no la variable de Laplace).

Una vez obtenida la representación, se utiliza, por ejemplo, para obtener un margen de estabilidad robusta mediante el teorema de pequeña ganancia. En este caso la estabilidad está garantizada para $|\delta(t)| \leq 0.46$, incluso aunque δ no sea constante (con δ constante podría calcularse estabilidad mediante fórmulas de valores propios, pero no con δ variante en el tiempo o no-lineal, como $\delta = x_1^3$).

[867: lfttr2]

Modelado LFT para Control Robusto: 5, multiples parámetros: introducción y ejemplo Matlab

*****  30:05

Materiales:

[[CÓD.:](#) electricallftMIMO.mlx] [[PDF](#)]

[YouTube ▶]

Este vídeo analiza el modelado de un proceso $\dot{x} = A(\delta_1, \delta_2, \dots)x + B(\delta_1, \delta_2, \dots)u$ donde las matrices del sistema sean funciones racionales de varios parámetros inciertos.

La metodología meramente se esboza: está bastante clara en modelos de primeros principios donde cada parámetro incierto aparezca una vez, pero no hay una metodología clara para realizaciones multidimensionales en modelos genéricos manipulados.


Como ejemplo, se propone modelar el circuito eléctrico con resistencia incierta modelado en el video [864], donde además el condensador es no-lineal, con una ecuación $C_1(v_C)\frac{dv_C}{dt} = i_C$, con $C_1(v_C) = 40 - 30 \log_{10}(v_C)$ pF.

El vídeo finaliza con conclusiones sobre todo el proceso de modelado descrito en los videos [863] a [867].

29.2 Sistemas no lineales: acotación de sector

[868: OA1ftnolin]

Modelado LFT de sistemas no-lineales de sector

***** 

10:47

Materiales: [ModelLFTNoLinealObjAprend.pdf]

[YouTube ▶]


Este vídeo detalla los pasos necesarios para incorporar no-linealidades $z = f(w)$ en el modelado LFT descrito en los videos [863] a [867]. *Nota:* como alternativa, el vídeo [869] presenta conceptos similares con un poco más de detalle.

- Concepto de no-linealidad acotada por sector.
- cambio de variable, a sector de pendiente ± 1 , esto es $\bar{f}(w)w \leq w^2$.
- Embebido LPV $\bar{f}(w) = \delta(t)w(t)$, para $\delta(t) = f(w)/w$, $|\delta| \leq 1$.
- Con ello, extrayendo $\delta(t)$ según explicado en los vídeos anteriormente referidos, se puede analizar estabilidad robusta (o diseñar controladores robustos) que toleren no-linealidad.
- Ejemplo numérico $\dot{x}_1 = -2x_1 + x_2$, $\dot{x}_2 = \sin(x_1 + x_2) - 2x_2$. Prueba de estabilidad por teorema de pequeña ganancia.

Otro ejemplo numérico sobre un modelo de depósito se detalla en el vídeo [871].

[869: lftnolin]

Modelado LFT incierto de sistemas no lineales para control robusto (versión extendida)

*****  15:33

Materiales:

[ModelLFTNoLinealObjAprend.pdf]

[PLAY ▶]

Este vídeo presenta los mismos contenidos que el vídeo [868], pero con algo más de extensión y detalle en la explicación de algunos de los conceptos. Dependiendo de las “prisas”, el estudiante sólo necesita visionar uno de los dos.

29.2.1 Ejemplos Matlab

Caso de estudio: depósito de primer orden

[870: dsn11]

Prestaciones robustas ante no linealidad, depósito 1er orden (I):

Modelado, linealización

Materiales: [[CÓD.:](#) SectorNonlinDeposito.mlx] [[PDF](#)]

** 10:27

[YouTube ►]

Este vídeo es el primero de una serie de cuatro vídeos que analizan el caso de estudio de control lineal robusto ante no-linealidades, sobre un ejemplo de un depósito de primer orden. Se ha escogido ese modelo sencillo para ilustrar el procedimiento, aunque realmente en una aplicación práctica con ese tipo de proceso sería “matar moscas a cañonazos”. Un ejemplo sobre un proceso de segundo orden aparece en el vídeo [874].

Este primer vídeo revisa el modelo, la linealización (derivadas parciales \times incrementos) y la transformación a coordenadas incrementales; todos son conceptos bien conocidos seguramente, pero sientan las bases y la notación para los siguientes vídeos; es posible que no necesites visionar este vídeo para comprender el resto del caso de estudio si los conceptos te son bien familiares, dado que las ideas de aquí están resumidas en el primer par de minutos del segundo vídeo.

También revisa la linealización por realimentación (cancelación de la raíz cuadrada), pero como eso da lugar a controladores **no** lineales, no va a ser objetivo del caso de estudio (se desean controladores *lineales*)... realmente, si se puede hacer linealización por realimentación, es posible que sea una muy buena opción si el modelo de la no-linealidad es razonablemente exacto, pero no en todos los procesos puede hacerse o/y pueden resultar controladores no lineales “demasiado complejos” que el cliente no esté dispuesto a implementar si existe un control lineal satisfactorio.

La linealización por derivadas parciales o la cancelación de no-linealidades (linealización por realimentación) dan lugar a modelos lineales *sin incertidumbre*. El caso de estudio continua en el vídeo [871], proponiendo, como alternativa, una linealización *de sector* que resulta en un modelo lineal *con incertidumbre*.

[871: dsn12]

Prestaciones robustas ante no linealidad, depósito 1er orden

(II): Acotación de no-linealidad en sector

Materiales: [[CÓD.:](#) SectorNonlinDeposito.mlx] [[PDF](#)]

*** 12:12

[YouTube ►]

Este vídeo es el segundo de una serie de cuatro vídeos que analizan el caso de estudio de control lineal robusto ante no-linealidades, sobre un ejemplo de un depósito de primer orden.

El vídeo [870] revisa el modelo, la linealización y la transformación a coordenadas incrementales. El resultado eran modelos lineales *sin incertidumbre*. Hasta el instante [02:30] de este vídeo, se hace un resumen rápido del vídeo referido.

Este vídeo prescinde de la linealización “clásica” por Taylor y, en su lugar, acota (en coordenadas incrementales) la no-linealidad entre dos rectas (acotación de sector). Las ideas teóricas y otros ejemplos se desarrollan en el vídeo [868].

Tras acotar en el sector, tomando la bisectriz como “modelo nominal”, se hace un cambio de variable de modo que el espacio entre las dos rectas se describe como una ganancia incierta que depende de cierto Δ que pertenece al intervalo $[-1, +1]$. El resultado de esta acotación es, por tanto, un modelo lineal *incierto* que “cubre” a la no-linealidad.

El caso de estudio continua en el vídeo [872], donde se utiliza la acotación calculada para plantear una planta generalizada con incertidumbre para un problema de seguimiento de referencia 1GL.

[872: dsnl3] **Prestaciones robustas ante no linealidad, depósito 1er orden (III): Planta generalizada no ponderada**

Materiales: [[CÓD.:](#) disenyoreguladordepositoSIM.zip] [[PDF](#)]

**** 10:59

[YouTube ▶]

Este vídeo es el tercero de una serie de cuatro vídeos que analizan el caso de estudio de control lineal robusto ante no-linealidades, sobre un ejemplo de un depósito de primer orden.

El vídeo [870] revisa el modelo, la linealización y la transformación a coordenadas incrementales; el vídeo [871] realiza la acotación de sector y el escalado de modo que la incertidumbre esté en $[-1, +1]$. Hasta el instante [02:45] de este vídeo, se hace un resumen rápido de los vídeos anteriores.

Aquí se plantea una planta generalizada con incertidumbre 3×3 para seguimiento de referencias, con un grado de libertad: aparte de entradas desde la incertidumbre y salidas a la misma, se incorporan error y acción de control como salidas cuya norma desea minimizarse, referencia como entrada exógena y la información al controlador (error) y su decisión (caudal).

El caso de estudio continua en el vídeo [873], donde se genera la planta generalizada ponderada y se diseñan controladores basándose en el teorema de pequeña ganancia escalado.

[873: dsnl4] **Prestaciones robustas ante no linealidad, depósito 1er orden (IV): planta ponderada, cálculo controlador, simulación**

Materiales: [[CÓD.:](#) disenyoreguladordepositoSIM.zip] [[PDF](#)]

***** 18:52

[YouTube ▶]

Este vídeo es el último de un caso de estudio de modelado y control robusto *lineal* para un proceso (depósito) no lineal. Hasta el instante [01:45] se hace un resumen rápido de los vídeos anteriores.

Con la planta generalizada no ponderada del vídeo [872], se codifican las prestaciones como pesos de error y acción de control (similar a los problemas tipo sensibilidad mixta: paso bajo en error, constante en acción de control).

Se resuelven tres problemas de control \mathcal{H}_∞ :

- Sólo prestaciones nominales (ancho de banda, saturación –límite de u –). Realmente, ese diseño resulta en estabilidad robusta ante incertidumbre aditiva, pero no es al menos literalmente la incertidumbre no lineal aquí estudiada.
- Prestaciones robustas sin multiplicador (teorema de pequeña ganancia no escalado). No sale factible la solución ($GAM > 1$).
- Prestaciones robustas con multiplicador. Sí sale factible $GAM < 1$, por lo que se prueban las prestaciones.

La teoría subyacente se discute en el vídeo [835]. Otro ejemplo con una rutina que busca el multiplicador en un bucle se discute en los vídeos [837] y [838]. Otro ejemplo parecido a lo que aquí se propone de sistema no-lineal acotado por sector y prestaciones robustas se desarrolla en el vídeo [874].

La parte final del vídeo realiza dos simulaciones del regulador resultante (que es grosso modo un Proporcional-Integral):

- Simulación lineal cerrando el bucle sustituyendo la incertidumbre por una constante (que toma valores entre -1 y +1). Se comprueban respuesta temporal y en frecuencia. La interpretación debe ser “informal” y “orientativa” dado que el proceso subyacente es no lineal y, por tanto, no tiene respuesta en frecuencia ni función de transferencia.
- Simulación no-lineal, en Simulink, donde se comprueban las prestaciones conseguidas al variar la altura. Son satisfactorias.

Como se dijo en los comentarios del primer vídeo de este caso de estudio, esto es “matar moscas a cañonazos”, dado que en un control de nivel de primer orden todo funciona bien, incluso un simple control todo-nada (con histéresis)... hemos presentado una forma complicada de sintonizar un PI que hubiésemos quizás podido hacer a mano. Pero el objetivo era comprender la metodología con un proceso muy sencillo para poder aplicarla a casos más complejos. El ejemplo del vídeo [874] (rechazo de perturbaciones en un sistema de segundo orden) estaría más cerca del ámbito donde la técnica propuesta sería más provechosa.

Ejemplos adicionales: sistema de segundo orden

[874: nlfml] Modelado de sistemas no-lineales con LFT: ejemplo Matlab

***** 17:34

Materiales: [[CÓD.](#): solrobnlcurt.mlx] [[PDF](#)]

[YouTube ▶]

Este vídeo presenta un ejemplo adicional sobre modelado de un sistema no lineal y diseño de controladores lineales robustos que toleren la no-linealidad de sector, mediante teorema de pequeña ganancia escalado (utilizando un multiplicador *constante*) para la no-linealidad. Las operaciones sobre los modelos, diagramas de bloques y diseño de controlador son implementadas en su totalidad en Matlab, mediante el uso de `lft` y `hinfsyn`. Por brevedad, se obtiene la respuesta ante escalón y en frecuencia de un bucle cerrado con incertidumbre Δ constante, aunque la simulación correcta debería haberse ejecutado mediante `simulink` o `ode45` dado que el sistema es no-lineal y, por tanto, presentará *distorsión armónica*, tal y como se ha hecho en el caso de estudio de un depósito del vídeo [873].

Otro caso de estudio de acotación de sector y prestaciones robustas ante no-linealidad en un modelo de depósito se discute en los vídeos [871] (modelado), [872] (planta generalizada) y [873] (aplicación tma. pequeña ganancia escalado).

29.3 Linealización alrededor de trayectorias

[875: ltray] Linealización de sistemas alrededor de trayectorias (teoría)

**** 10:59

Materiales: [LinealizTray.pdf]

[YouTube ▶]


En este vídeo se generaliza la linealización de una ecuación de estado $\dot{x} = f(x, u)$, que en el capítulo 3 se linealiza alrededor de un punto de equilibrio constante (ver vídeo [86]), resultando en variables incrementales $\frac{d\Delta x}{dt} = A \cdot \Delta x + B \cdot \Delta u$.

El nuevo problema estudiado aquí es el de linealizar alrededor de una *trayectoria* (x_{ref}, u_{ref}) , que verifique $\dot{x}_{ref} = f(x_{ref}, u_{ref})$. El resultado es un sistema lineal *variante en el tiempo* $\frac{d\Delta x}{dt} = A(t) \cdot \Delta x + B(t) \cdot \Delta u$ cuya estabilidad o diseño de controladores podrían ser abordados mediante técnicas de control robusto. Obviamente, el caso de un equilibrio constante es un caso particular de este proceso más general.

Las trayectorias pueden ser originadas por “demostración” (para luego intentar repetir un experimento que salió “perfecto”), o por “inversión dinámica” (por ejemplo, calcular pares para que un sistema tenga unas aceleraciones deseadas), aunque en el segundo caso suelen existir otras técnicas directamente no lineales quizás más recomendables para diseño de control (en particular la linealización exacta por realimentación cancelando no-linealidades, flatness, ...).

[876: ltraym]

Linealización de sistemas alrededor de trayectorias (ejemplo Matlab)


19:54

Materiales:

[Cód.: lintinvpesim.mlx]

[PDF]


[YouTube ▶]

Este vídeo compara las simulaciones no-lineal, linealizada en un punto de equilibrio y linealizada en una trayectoria de un modelo de péndulo de segundo orden (teoría en vídeo [875]).

La linealización alrededor de un punto de equilibrio constante (dando como resultado un sistema lineal invariante en el tiempo) estaba justificada para calcular la respuesta mediante transformada de Laplace, exponenciales, etc. La linealización *variante* en el tiempo (trayectoria) debe, en general, ser integrada numéricamente; por ello, no debe ser considerada como una herramienta de cálculo de respuesta o simulación, sino por su interés conceptual en sí mismo y para futuras pruebas teóricas de estabilidad en bucle abierto o bucle cerrado con, por ejemplo, técnicas de control robusto o planificación de ganancia.

[877: ltr2]

Linealización alrededor de trayectorias: consideraciones adicionales y relaciones con otros enfoques


13:19

Materiales: [LinealizTray2.pdf]

[YouTube ▶]

En este vídeo se discuten más detalles de la linealización alrededor de una trayectoria: en concreto, su relación con el control (lineal) con dos grados de libertad, que se discute en el vídeo [519], su relación con el filtro de Kalman extendido (problemas de estimación del estado, vídeo [616]), y las posibilidades teóricas de analizar el control del sistema variante en el tiempo $\dot{x} = A(t)x + B(t)u$ (incremental) resultante:

1. Control lineal robusto (controlador invariante en el tiempo que tolere la incertidumbre).
2. Control por planificación de ganancia (que cambie sus parámetros según cambian los coeficientes de $A(t)$ y $B(t)$, por ejemplo).
3. Control predictivo variante en el tiempo que utilice la referencia futura y los valores de A y B en instantes futuros.

La tercera de las opciones sería, obviamente, la mejor de todas.

Capítulo 30


Pasividad

*Este capítulo describe brevemente y desde un punto de vista intuitivo el concepto de pasividad (disipación de energía), aunque con excesivas “licencias” desde el punto de vista formal. Se recomienda familiarizarse con los conceptos de estos materiales, pero posteriormente no pasar por alto la lectura en detalle de las definiciones y teoremas en libros de control no lineal: no confiar *textualmente* en los teoremas descritos en este capítulo dado que contienen “incorrecciones”, por simplicidad y brevedad de exposición.

30.1 Definiciones básicas y utilidad

[878: pas0] Pasividad: definición y ejemplos estáticos sencillos

Materiales: [Pasividad0defs.pdf]

***  13:18
[YouTube ►]

Este vídeo define los sistemas pasivos en forma integral, como aquéllos en los que $\int_0^T u^T y \, dt \geq 0$, y en forma de función de almacenamiento $u^T y \geq \frac{dV(x)}{dt}$.

Se discuten las relaciones del concepto de “pasividad” en teoría de control con el concepto de “sistema conservativo” en Física, y de “pasividad estricta” con sistema disipativo.

El vídeo pone ejemplos de sistemas estáticos (resistencia, rozamiento, tubería, no-linealidades de primer y tercer cuadrante) donde se cumple la propiedad de pasividad y lanza la idea intuitiva que sin aporte de potencia externo, la energía disminuirá en un sistema disipativo, lo cual bajo ciertas condiciones dará lugar al “teorema de pasividad” para probar estabilidad, que no entra en los objetivos de este material (ver vídeo [879]).

[879: pas1] Pasividad y estabilidad, interconexión de sistemas

Materiales: [Pasividad1bTMA.pdf]

****  11:56
[YouTube ►]

Este vídeo presenta la relación entre pasividad estricta (vídeo [878]) y estabilidad, donde la propia función $V(x)$ de almacenamiento de energía puede considerarse, bajo ciertas suposiciones, una función de Lyapunov para probar estabilidad de sistemas tanto lineales como no lineales.

También discute la pasividad (y, por tanto, si es estricta y se cumplen otras condiciones de observabilidad, la estabilidad) de sistemas pasivos interconectados con realimentación positiva o negativa, y la relación de la segunda con la mera conexión eléctrica o mecánica en un sistema físico real.

Nota: la presentación está hecha de modo muy “informal” con poco rigor, centrándose en los aspectos ‘intuitivos’ que motivaron este campo: se remite al lector a libros de texto como Green & Limebeer (Linear Robust Control) o Khalil (Nonlinear Systems) para un tratamiento riguroso de estos conceptos.

En control robusto, muchas especificaciones se establecen como que determinadas normas (planta generalizada ponderada) sean menor a 1. Integrar pasividad junto a otras consideraciones de robustez, todo en un enfoque unificado, requiere transformaciones pasivo↔contractivo, consideradas en los vídeos [888] y [889].

30.1.1 Refinamientos, exceso de pasividad

[880: paskh]

Pasividad y exceso de pasividad en sistemas dinámicos, estabilidad

10:59

Materiales:

[Pasividad1aSistemasDin.pdf]

[YouTube ▶]

Este video detalla un poco mejor las nociones preliminares sobre pasividad del vídeo [878] en el caso general de sistemas dinámicos, así como los teoremas de estabilidad, esbozados en el vídeo [879] para los casos de pasividad estricta o de pasividad output estricta + observabilidad del estado cero. Los detalles y justificación formal de todo esto deberían ser consultados en el libro *Nonlinear Systems* de H. Khalil, por ejemplo.

El “exceso de pasividad” puede ser, en ciertos casos, entendido como un modelado físico (resistencia mínima, conductancia mínima, ...). Este enfoque se aborda en los vídeos [885] y [886].

[881: paskhrc]

Pasividad y exceso de pasividad en sistemas dinámicos: balances de potencias, ejemplo circuito RC serie

14:30

Materiales:

[Pasividad1aSistemasDin.pdf]

[YouTube ▶]

Este vídeo plantea cómo determinar si existe un exceso de pasividad (según discutido en el vídeo [880]) a partir de un balance de potencias en un sistema físico: *potencia inyectada* = *potencia almacenada* + *potencia disipada*.

Minimizando la expresión de la potencia disipada respecto a distintos argumentos puede demostrarse exceso de pasividad (pasividad estricta) estado/input/output. En muchos casos, los modelos físicos permiten establecer una resistencia, rozamiento, etc. mínimo para determinar el exceso de pasividad; este enfoque se aborda mediante ejemplos en el vídeo [885], por ejemplo.

Como ejemplo, se aplica a un ejemplo de circuito RC serie considerando distintas entradas y salidas:

1. Conductancia (entrada tensión, salida intensidad)
2. Impedancia (entrada intensidad, salida tensión)
3. Filtro (entrada tensión, salida tensión en condensador)

se determina si es o no input/output/estado estrictamente pasivo en cada uno de los tres casos.

El exceso de pasividad a partir de la respuesta en frecuencia se aborda en el vídeo [883] y ejemplos Matlab aparecen en el vídeo [884].

30.1.2 Sistemas lineales pasivos: condiciones respuesta frecuencia

[882: pasltifase] Pasividad en sistemas lineales (1): condiciones de fase (SISO) *** 06:19
Materiales: [PositiveRealTF.pdf] [YouTube ▶]

Este vídeo motiva la necesidad de determinar la pasividad de un sistema lineal invariante en el tiempo arbitrario, si no se dispone de un modelo físico del que extraer la función que determina la energía almacenada en función del estado, para comprobar pasividad de controladores, filtros, etc.

Existen métodos de búsqueda numérica de la “energía” $x^T Px$, y existen condiciones sobre la respuesta en frecuencia para determinar la pasividad.

El objetivo es argumentar por qué en el caso SISO la condición de pasividad es que el desfase (argumento de la resp. en frecuencia, α) verifique $-90^\circ \leq \alpha \leq +90^\circ$, a partir de la integral $\int \sin(\omega t) \sin(\omega t + \alpha) dt$.

En el vídeo [883] se aborda el caso general multivariable, al cual el concepto de “fase” no se puede aplicar.

[883: pasltispr] Pasividad sistemas lineales: matrices de transferencia (strictly) positive real ***** 10:34
Materiales: [PositiveRealTF.pdf] [YouTube ▶]

Este vídeo generaliza las ideas preliminares sobre la fase $-90^\circ \leq \alpha \leq +90^\circ$ de la respuesta en frecuencia de sistemas pasivos (vídeo [882]), al caso multivariable donde la fase no aplica.

Básicamente, dado un sistema $\Delta(s)$ se justifican mediante Parseval (integral de productos en tiempo = integral de productos en transformada de Fourier –frecuencia–) condiciones sobre el componente hermitico $\frac{1}{2}(\Delta(j\omega) + \Delta^*(j\omega))$, que debe ser semidefinido positivo en el caso pasivo y definido positivo en el caso estrictamente input pasivo; o las condiciones sobre la inversa de Δ para el caso estrictamente output pasivo (ver definiciones en vídeo [880]).

Estas justificaciones, realmente, son “informales”: el vídeo enuncia sin demostrar la equivalencia entre una matriz de transferencia “positive real” y pasividad (y las condiciones que deben cumplir dichas matrices de transferencia) así como pasividad estricta \Leftrightarrow matriz de transferencia “strictly positive real”.

Ejemplos Matlab de estas condiciones aparecen en el vídeo [884].

El exceso de pasividad, en bucles realimentados, permite un “defecto de pasividad” en los componentes interconectados. En el ámbito de la frecuencia, eso se plasma en el bien conocido “criterio del círculo”, ver vídeo [887].

[884: sprml] Pasividad y exceso de pasividad SISO: respuesta en frecuencia (ejemplo Matlab) ***** 06:09
Materiales: [Cód.: PASejemplosLineales.mlx] [PDF] [YouTube ▶]

Este vídeo presenta un ejemplo Matlab donde se analizan las características de la respuesta en frecuencia (diagramas de Nyquist y Bode) de sistemas lineales pasivos, estrictamente input pasivos, estrictamente output pasivos y no pasivos. La teoría sobre dichas condiciones aparece en los vídeos [882] (caso SISO, condiciones de fase) y [883] (caso general).

Se utilizan los comandos Matlab `isPassive` y `getPassiveIndex` explicando su funcionamiento y significado.

30.1.3 Exceso de pasividad, criterio del círculo (Matlab)

[885: pasexc1] Exceso de pasividad (1.-input, 2.-output): ejemplo LFT Matlab

***  14:45

Materiales: [[CÓD.:](#) PasivLFTejemplos1.mlx] [[PDF](#)]

[[YouTube](#) ►]

Este vídeo presenta ejemplos de lo que se suele denominar “exceso” de pasividad: una configuración donde no sólo un sistema es pasivo sino que existen resistencias (conocidas) adicionales, o rozamientos mínimos, etc.

El caso general es una interconexión $lft(G, \tilde{z})$ siendo $\tilde{z} = lft(z, F)$ donde z es pasivo y F es un subsistema estático (matriz constante conocida) que modela esa disipación mínima conocida. A partir del hecho de que $-H$ debe ser pasivo, siendo $H = lft(F, G)$, se puede (invirtiendo las ecuaciones que representan dicha lft) determinar la zona del plano complejo donde la respuesta en frecuencia de G debe tomar valores, dado que sabemos que H debe estar en el semiplano izquierdo del plano complejo.

En concreto, se presenta un ejemplo Matlab donde se calcula la interconexión F y la región donde $G(j\omega)$ debe tomar valores para dos casos:


1. Resistencia (conocida) en serie con un circuito pasivo desconocido, cuya disipación mínima será $R \int i^2 dt$;
2. Resistencia (conocida) en paralelo con un circuito pasivo desconocido, cuya disipación mínima será $\frac{1}{R} \int V^2 dt$.

Una discusión adicional sobre la interrelación entre ambos casos y analogías mecánicas se aborda en el vídeo [886]. La combinación de ambos en una configuración serie+paralelo con dos resistencias conocidas se aborda en el vídeo [887], donde se presenta el popular “criterio del círculo”.

Un caso adicional (circuito RC con fugas) también traducido a una LFT con un circuito pasivo se aborda en el vídeo [891], donde se discute el modelado, y su inclusión en una planta generalizada para μ -síntesis en el vídeo [892]. Este enfoque es, realmente, el más general y válido para el caso multivariable, teniendo los dos casos explicados en este vídeo importancia tutorial e histórica, pero que sólo aplican al caso monovariable.

El exceso de pasividad también puede ser determinado a partir de balances de potencias sin conocer al detalle la dinámica completa, según se discute en el vídeo [881].

[886: pasexc2] Exceso pasividad input/output: analogía mecánica, relación entre ambos casos

***  07:57

Materiales: [[CÓD.:](#) PasivLFTejemplos2.mlx] [[PDF](#)]

[[YouTube](#) ►]

En este vídeo se ahonda en los aspectos físicos de modelado de los casos serie o paralelo discutidos en el vídeo [885]. En concreto, se abordan analogías mecánicas (amortiguadores en serie o paralelo con sistemas mecánicos pasivos) con las mismas ecuaciones que los ejemplos eléctricos del citado vídeo.

También se esboza la idea de que, en realidad, los dos casos son el mismo: intercambiando los roles de entrada y salida entre intensidades y tensiones, los diagramas de bloques del caso 1 se convierten en el caso 2 y viceversa.

Una conexión combinada serie+paralelo y su analogía mecánica se discuten en el vídeo [887] que da lugar al llamado “criterio del círculo”.

[887: pascricir] Pasividad (exceso input+output, serie+paralelo): criterio del círculo

****  13:26

Materiales: [[CÓD.:](#) CritCircLFTstatic.mlx] [[PDF](#)]

[[YouTube](#) ►]

En este vídeo se generalizan las interconexiones discutidas en los vídeos [885] y [886], abordando el caso de un circuito eléctrico con un componente R_1 en serie con dos elementos en paralelo (una resistencia R_2 conocida y un circuito pasivo de impedancia z desconocida). En este circuito se tiene, por tanto, una impedancia resistiva mínima de R_1 si z es un cortocircuito, y una impedancia máxima de $R_1 + R_2$ si z es un circuito abierto.

También se presenta un análogo mecánico de dicho sistema eléctrico, conectando en serie y paralelo amortiguadores con otros elementos mecánicos pasivos (cualquier conjunto arbitrario de masas, muelles, amortiguadores o incluso no-linealidades pasivas).

Como resultado de las manipulaciones se llega al famoso *criterio del círculo*, que es el resultado más importante de pasividad de un componente lineal $G(s)$ con otros componentes posiblemente no lineales de ganancia en un intervalo: si la respuesta en frecuencia $G(j\omega)$ no entra en el círculo cuyo diámetro es el segmento $[1/R_{\max}, 1/R_{\min}]$ y da las vueltas adecuadas (Nyquist) a dicho círculo según el número de polos inestables en bucle abierto de G , entonces la interconexión $\text{lft}(z, \text{lft}(F, G))$ es estable.

El resultado presentado es monovariable. Intencionalmente, se presenta de forma bastante diferente a mucha de la literatura de control no lineal, bajo el formalismo "lft" porque el objetivo global es entender cómo incorporar pasividad en problemas de control robusto y planta generalizada: un caso relativamente similar (circuito RC con fugas) traducido a una LFT con un circuito pasivo se modela en el vídeo [891], y se incluye en una planta generalizada para μ -síntesis en el vídeo [892]. Este enfoque es, realmente, el más general y válido para el caso multivariable; el criterio del círculo tiene relevancia histórica y como metodología para, quizás análisis o diseño monovariable de reguladores sencillos, pero el enfoque de integrar la lft F en una planta generalizada es más moderno y general.

30.2 Pasividad para control robusto \mathcal{H}_∞

30.2.1 Transformaciones pasivo-contractivo

[888: pasc1] Transformación pasivo-contractivo (parte 1)

Materiales: [Pasividad2TransfContractivoPart1.pdf]

****  12:56

[YouTube ►]

Es este material se discute la transformación entre un sistema pasivo D que verifique $\int u^T y > 0$ y uno contractivo \hat{D} que verifique $\int m^T m < \int n^T n$ y, por tanto $\|D\|_\infty < 1$ (o la norma inducida $2 \rightarrow 2$ si es no lineal). con $m = y - u$, $n = y + u$. Las definiciones básicas de pasividad se discuten en el vídeo [878], y la relación con la estabilidad en el vídeo [879].

Se requieren ciertas condiciones de pasividad *incremental* para que en un caso no-lineal determinadas operaciones estén bien planteadas (invertibles). También se considera el caso $\int u^T y < 0$ que debe usarse en bucles con realimentación positiva (se resuelve intercambiando m y n o, equivalentemente, cambiando el signo de D).

La parte final del vídeo expresa las transformaciones pasivo \rightarrow contractivo en forma de diagrama de bloques, y también en forma LFT, conveniente para la interconexión de sistemas complejos donde intervengan estas transformaciones.

La transformación inversa y su uso en bucles de control se discuten en el vídeo [889], continuación de éste.

[889: pasc2] Transformación pasivo-contractivo (parte 2)

Materiales: [Pasividad2TransfContractivoPart2.pdf]

****  11:57

[YouTube ►]

Este vídeo es continuación del vídeo [888] donde se introdujo la transformación pasivo-contractivo. En este vídeo se comprueba que la transformación inversa convierte un sistema contractivo en uno pasivo $\int u^T y > 0$ y, cambiando ciertos signos, en uno negado pasivo $\int u^T y < 0$. También se presenta la expresión de dicha transformación en forma LFT.

En la segunda parte de este vídeo (minuto [06:39]), se aborda el hecho que la combinación de ambas transformaciones produce la identidad. Ese hecho puede aprovecharse en un diagrama de bloques para convertir un bucle $G-\Delta$ cuya estabilidad se prueba por pasividad (vídeo [879]), en otro bucle donde interviene el modelo transformado \tilde{G} de modo que su estabilidad pueda probarse mediante teorema de pequeña ganancia (vídeo [798]), ilustrando la relación teórica entre ambos conceptos (ambos son un caso particular de lo que se denomina *integral quadratic constraints*, fuera de los objetivos de este material).

La Robust Control Toolbox de Matlab incorpora estas transformaciones de forma transparente al usuario, tal y como se discute en el caso de estudio en el vídeo [892].

[890: pasctml] Transformación pasivo-contractivo (ejemplo Matlab) **** 08:38
Materiales: [Cód.: transformpasicontract.mlx] [PDF] [YouTube ►]

Este vídeo ilustra las transformaciones pasivo→contractivo (vídeo [888]) y su inversa (vídeo [889]) así como su interconexión, comprobando cómo el semiplano derecho del dominio de Laplace se transforma en el círculo unidad (o el izquierdo, necesario en conexiones con realimentación positiva) en un caso, o viceversa, el círculo se transforma en un semiplano. Se comprueba el diagrama de Nyquist de varios procesos. Se comprueba el resultado del comando Matlab `getPassiveIndex`.

30.2.2 Caso de estudio (planta generalizada)

[891: paselml1] Modelado de un circuito RC con impedancia de fugas pasiva *** 08:00
Materiales: [Cód.: EjemploElectricoMuSyn.mlx] [PDF] [YouTube ►]

En este vídeo se modela un circuito RC serie donde también se supone que existe una impedancia de fugas que se puede expresar como un componente resistivo R_f y otro componente desconocido pasivo de impedancia $z_p(s)$.

Para poder modelarlo de forma que se obtenga una interconexión LFT entre una planta generalizada y un bloque incierto Δ se ha sustituido la impedancia incierta por una fuente de tensión ideal.

En principio, aquí no se usa el hecho de que la incertidumbre sea o no pasiva: en control, eso se usará en el problema de μ -síntesis discutido en el vídeo [892], continuación de éste.

La parte final del vídeo comprueba la calidad del modelo obteniendo la respuesta temporal y en frecuencia de distintas impedancias z_p : circuito abierto ($z_p = \infty$), cortocircuito ($z_p = 0$), inductiva ($z_p = Ls$) o capacitiva ($z_p = \frac{1}{Cs}$).

[892: paselml2] Mu-síntesis con incertidumbre pasiva: caso circuito RC con fugas **** 12:59
Materiales: [Cód.: EjemploElectricoMuSyn.mlx] [PDF] [YouTube ►]

Este vídeo utiliza el modelo incierto de un circuito RC con impedancia de fugas pasiva visto en el vídeo [891]. En concreto, usa la opción `PositiveReal` del bloque `ultidyn` de la Robust Control Toolbox.

Conectando dicha `ultidyn` en LFT con el modelo del vídeo [891], se obtiene una planta incierta que puede integrarse en una planta generalizada, planta generalizada ponderada y usarse en `musyn`.

La Robust Control Toolbox obtiene de forma transparente al usuario los modelos lft con la incertidumbre desconocida “fuera” de la planta generalizada, así como las transformaciones pasivo→contractivo necesarias en este caso (vídeos [888] y [889]). Por tanto, no es necesario conocer los detalles técnicos o teóricos (vídeo [879]) para usar de forma muy sencilla las incertidumbres pasivas en problemas de control robusto gracias a la toolbox de Matlab.

Capítulo 31

El enfoque LMI a control robusto y planificación de ganancia (DRAFT)

Desigualdades Matriciales Lineales

Las desigualdades matriciales lineales (linear matrix inequalities, LMI) son un tipo de restricciones y problemas de optimización convexa asociados, que datan de alrededor de 1990, que tienen un papel fundamental en problemas geométricos, diseño de controladores y incluso análisis de datos multivariantes.

Las desigualdades matriciales lineales se han convertido, a partir de mediados de los años 1990, en una herramienta numérica capaz de obtener soluciones a problemas complejos de control óptimo y robusto (de hecho, los problemas LQR, \mathcal{H}_2 y \mathcal{H}_∞ pueden ser formulados como LMIs).

El desarrollo completo de estas ideas daría lugar a un libro completo... el libro de texto seminal en gran parte de estos aspectos fue <https://web.stanford.edu/~boyd/lmibook/lmibook.pdf>, cuya lectura se recomienda a estudiantes avanzados/Doctorado (su nivel matemático puede ser excesivo para perfiles de Grado o Máster en ingeniería). También el libro <https://www.wiley.com/en-es/Fuzzy+Control+Systems+Design+and+Analysis:+A+Linear+Matrix+Inequality+Approach-p-9780471465225> es una referencia especializada en las aplicaciones a control de las desigualdades matriciales lineales desarrolladas en los años 1990; se basa en el libro de Boyd, pero es, quizás más accesible (o al menos tiene más ejemplos y un lenguaje algo menos formal).

Aquí incluyo desarrollos y ejemplos Matlab que cubren parcialmente determinados aspectos para los que, por una razón u otra, sí que tengo código o material teórico elaborado.

31.1 Problemas geométricos resolubles con LMIs

31.1.1 Condicionamiento mínimo

Esta sección presenta un ejemplo de aplicación a condicionamiento de plantas, usando el solver LMI-Lab de la Robust Control Toolbox de Matlab.

[893: mincond]

Minimización de número de condición ante escalado diagonal
(teoría)

***** ⌂ 15:49

Materiales: [mincond.zip]

[PLAY ▶]

En este vídeo se define el número de condición mínimo ante escalado diagonal, $\min_{E_y, E_u} \gamma(E_y^{-1} M E_u)$, ampliando los conceptos básicos discutidos en el vídeo [786] sobre condicionamiento numérico. La matriz de ganancia relativa (RGA) también tiene cierta relación con este número de condición mínimo (proporciona una cota inferior): de modo opcional, el lector interesado podría visionar la parte final del vídeo [472] o el ejemplo Matlab en el [473].

Una vez definido y planteado el problema, se expresa en forma de LMI –desigualdades matriciales lineales– (formalmente, como un problema de valor propio generalizado [gevp](#)). Se remite a código adjunto para su resolución numérica usando las herramientas LMI del Robust Control Toolbox de Matlab.


El vídeo [unfoldm(14:55)] también aplica este tipo de desigualdades al problema de “desdoble isométrico” en análisis de datos multidimensionales.

31.1.2 Geometría de elipses y elipsoides

[894: lmielout1EN]
18:43

LMIs: Ellipsoid containing other ellipsoids/polyhedra,
Yalmip/Sedumi/Matlab (1): minimum major axis

Materiales: [[CÓD.:](#) ElipsLMIout.mlx] [[PDF](#)]

**** 

[YouTube ▶]

This video presents how to obtain, via linear matrix inequalities (LMI) or *semidefinite programming* (SDP), according to other texts, the “smallest” ellipse that contains a given polygon and another ellipse.

The approach can be generalized to more ellipses and more polyhedra in more dimensions; the LMI code would be practically identical.

The first part of the video discusses the conditions for an ellipse $x^T P x \leq 1$ to contain a polyhedron with vertices v_i (that is, $v_i^T P v_i \leq 1$) and to contain another ellipse $x^T M x \leq 1$, that is stated as $M - P \succeq 0$; The second case is detailed and proven in the video [??].

The second part of the video discusses what needs to be “optimized” (objective function): in this first video, it is decided to minimize the length of the semimajor axis of the ellipse, that is, we wish to minimize the radius of the smallest sphere that contains the sought ellipse. This is done by minimizing ρ , adding the constraint $P - \rho^{-2} I \succeq 0$. Actually, to make the objective function and constraints linear in the decision variables, we change to maximize β subject to $P - \beta I \succeq 0$.


The video details all the YALMIP code that must be entered to solve the problem in a Matlab environment.

Note: by minimizing the size of the semimajor axis of the ellipse, the minor semi-axis is “free” and the solution is not unique. Discussing this topic as well as the minimization of the volume (area in a 2D case) of the desired ellipsoid is the objective of the video [895], a continuation of this one.

[895: lmielout2EN]
09:39

LMIs: Ellipsoid containing other ellipsoids/polyhedra,
Yalmip/Sedumi/Matlab (2): minimum area

Materiales: [[CÓD.:](#) ElipsLMIout.mlx] [[PDF](#)]

**** 

[YouTube ▶]

This video is a continuation of the video [894]. In said video, the LMI restrictions were proposed so that an ellipse included a polyhedron and another given ellipse. Then the radius of the circumscribed circumference of said ellipse was minimized, thereby minimizing the semimajor axis of the searched ellipse.


In this video it is discussed that, since the minor axis of the result is not unique, you can try to minimize it with 10,000 times less “force”... adding $1e-4$ times the trace of the matrix to the objective function, you get something similar to the ellipse with “minimum minor axis” among the many ellipses with “minimum major axis” feasible as an optimal solution to the problem we are considering.

The second part of the video discusses how to truly obtain the minimum area ellipse, using the Yalmip `geomean` command. This would obtain the minimum-volume ellipsoid in three or more dimensions.

[896: lmielin1EN]
16:49

LMI demos: largest ellipse inside other poliedra/ellipses (1)

Materiales: [[CÓD.:](#) ElipsLMIlin.mlx] [[PDF](#)]

**** 

[YouTube ▶]

This video discusses how to obtain using linear matrix inequalities (LMI) the largest ellipse within a polygon (well, it also supports sides that are an ellipse segment).

The presentation is divided into two videos.

In this first video the problem is posed, and we consider how to obtain the matrix P such that $x^T P x \leq 1$ is within the predetermined intersection set of polygons/ellipses.

The restrictions are coded as LMIs, and two possible cost indices are proposed:

- Minimize maximum eigenvalue of P , which is equivalent to maximizing semi-minor axis, that is, finding the circle with the largest radius contained in the search region.
- Minimize the trace of P ... minimize the sum of the inverse squares of the lengths of the semi-axes... This gives a 'large' ellipse as a result of the optimization but its meaning is not entirely clear. Indeed, searching for the ellipse with the largest area will be the objective of the video [897], a continuation of this one.

[897: lmielin2EN]
16:18

LMI demos: largest ellipse inside other poliedra/ellipses (2),
largest area (geomean)

***** 

Materiales: [[CÓD.:](#) ElipsLMIlin.mlx] [[PDF](#)]

[YouTube ▶]

This video discusses how to obtain, using linear matrix inequalities (LMI), the largest ellipse within a polygon (well, it also supports curved sides that are an ellipse segment). It is a continuation of [896] where the problem was posed and some possibilities were discussed with an ellipse in the form $x^T P x \leq 1$, P being a decision variable.

In this second video, Q is considered a decision variable that defines the ellipse $x^T Q^{-1} x \leq 1$. The 'inverse' form is used for convenience to express the optimization problem as a convex one. Indeed, *maximizing* the determinant of Q (well, actually its square root, geometric mean of the eigenvalues $\sqrt{\lambda_1 \lambda_2}$) can be done with the `geomean` operator of Yalmip.

Conveniently, the area of the ellipse is proportional to said geometric mean, so that we can solve the requested problem of maximum area. Furthermore, translating the restrictions to inverse form will require using 'congruence' and 'Schur's complement' so that the video will serve to illustrate the application of these results, which are very useful in control theory developments with LMIs.

[898: elinnccEN]

Maximum volume ellipsoid inside polyhedron and other
ellipsoids: 2D example, Matlab (LMI/SDP)

*****  15:36

Materiales: [[CÓD.:](#) ElipINelipORPOly.mlx] [[PDF](#)]

[YouTube ▶]


This video discusses the problem of obtaining the maximum volume ellipsoid inside a convex body defined as the intersection of polyhedra and ellipsoidal sets, not centered neither symmetric with respect to the origin as in other prior videos of the collection; well, as we are detailing a 2D example, we will actually speak about polygons and maximum area ellipse, but the ideas efficiently generalise to higher dimensions in a trivial way.

We use Matlab+YALMIP+SeDuMi to solve the associated semidefinite programming, setting linear matrix inequalities and a 'geomean' objective function (used by Yalmip/SeDuMi instead of 'logdet').

31.1.3 Conjuntos representables LMI/SDP

[899: lmisets1EN] LMI sets (SDP-representable sets, spectrahedra): definition, basic properties and 2D examples

17:42

**** 

Materiales: [[CÓD.:](#) LMIsExamplesPart1.mlx] [[PDF](#)]

[YouTube ►]

This video defines an ‘LMI set’, also known as ‘SDP-representable’ set, as the set of feasible values of a linear matrix inequality $LMI(x) \succeq 0$, where $\succeq 0$ means being positive semidefinite. The name of ‘spectrahedron’ (plural spectrahedra) is, too, used to denote these sets of feasible values of a semidefinite program.


Convexity of the cone of positive semidefinite matrices implies that LMI sets are convex sets. As positive-semidefiniteness of a matrix is equivalent to all principal minors being non-negative, then, LMI sets are semialgebraic ones with polynomial boundary.

Examples of LMI sets are given, such as circles, ellipses, polyhedra, cones and a cubic egg-shaped set. All examples are in 2D, for ease of visualization, albeit of course LMI sets can be defined in any dimension.

The next video [900] will describe examples of the so-called ‘lifted’ LMI sets, projections on a reduced dimension of LMI sets in a higher-dimensional space.

[900: lmisets2EN] Lifted LMI sets, 2D examples (Matlab, YALMIP)

13:56

**** 

Materiales: [[CÓD.:](#) LMIsExamplesParts1and2.mlx] [[PDF](#)]

[YouTube ►]

This video presents examples of ‘lifted’ LMI sets (SDP-representable sets), i.e., projections of higher-dimensional LMI sets, involving additional ‘dummy’ decision variables. This video is a continuation of video [899], which introduced the concept of LMI set, its convexity and polynomial boundaries, and posed some non-lifted examples. The first three minutes of the present video review such concepts.


Then, several examples of lifted sets are posed:

- a square with rounded edges, a 4D LMI set projected in 2D,
- an ellipse defined by its foci, also a 4D to 2D lifted set.
- a 3-ellipse (guitar pick), generalising from 2 to 3 the number of foci, so it is represented as a 2D projection of a 5D LMI set,‘
- a generic, arbitrarily invented 4×4 LMI in 5 variables, giving a convex shape with 4th-degree polynomial boundary (well, an extra linear constraint is added so degree might be considered to be 5) in the higher dimension.

Video [901] will discuss the details on plotting code, plus geometric and optimization problems that may be solved with these sets.

[901: lmisets3EN] LMI sets (3): plotting routine and applications

09:54

**** 

Materiales: [[CÓD.:](#) LMIsExamplesParts1and2.mlx] [[PDF](#)]

[YouTube ►]

This third video closes the introduction to LMI sets (and lifter ones). In here, details of the plotting code used in previous videos [899] and [900] is presented (it’s based on a ray-tracing idea, using YALMIP *optimizer* command, which is a faster pre-compiled version of *optimize* for repeated problems). Computation of the analytic center of an LMI set is also discussed.

The last part of the video discusses the kind of problems that can be addressed with SDP optimization software on LMI sets:


- Checking for intersection of several LMI sets (collision detection, say, in a planning problem)
- Optimization of linear functions on them (computation of tangent planes)
- Optimization of SDP-representable convex functions (as an example, the minimum distance between LMI sets; an explicit example of that is addressed in video [904] so it is omitted here).

Determining if one LMI set is contained into another may require, in a general case, other tools providing sufficient conditions via *Sum of Squares* (SOS) and Positivstellensatz multipliers; details on these extensions to LMIs is out of the scope of this introduction.

A later video with discussion on scaling, perspective cones and convex hull of LMI sets has been added, see [902].

[902: lmisets4EN] LMI sets (4): scaling, perspective cones, set interpolation, convex hull

20:15

***** 

Materiales: [[CÓD.:](#) LMIsetExamplesPart3Scaling.mlx] [[PDF](#)]

[YouTube ►]

This video discusses some transformations and operations with LMI sets (SDP-representable sets), introduced in videos [899] and [900].

The basic ideas are the following two ones:


1. Even if we name LMIs as ‘Linear’ matrix inequalities, they are actually ‘affine’, with a constant term $LMI(0)$ plus a truly linear one $LMI_L(x)$ so $LMI_L(ax + by) = a \cdot LMI(x) + b \cdot LMI(y)$. The linear component is, henceforth, $LMI_L(x) = LMI(x) - LMI(0)$.
2. if x is in the set such that $LMI(x) \succeq 0$, the transformation $z = Tx + q$ converts it to an LMI set in variable z given by $LMI(T^{-1}(z - q)) \succeq 0$.

From these two ideas, in this video we discuss:

- Scaling an LMI set
- Forming the ‘perspective cone’ in one more dimension, so the sections are the LMI set, whose projection is the union of all scalings in an interval
- Changing the apex (scaling point)
- Convex hull of LMI sets
- Convex interpolation of LMI sets (direct sum of λ and $(1 - \lambda)$ scaled versions).

31.1.4 Distancia Mínima entre elipses (o, en general, LMI sets)

[903: distelliEN] Distance between ellipses: semidefinite programming (SDP/LMI, linear matrix inequalities)

****  10:59

Materiales: [[CÓD.:](#) DistanceEllipsesENG.mlx] [[PDF](#)]

[YouTube ►]

This video explains how to compute the distance between two ellipses (minimum distance between any pair of points in them), by optimizing a function subject to linear matrix inequalities (LMI), that is, semidefinite programming (SDP).


The basic idea is to express the ellipses $(x_i - c_i)^T Q_i^{-1} (x_i - c_i) \leq 1$, $i \in \{1, 2\}$, as LMI sets, using Schur’s formula. Also a distance bound $(x_1 - x_2)^T (x_1 - x_2) \leq d^2$ will be expressed as an LMI on decision variables x_1 and x_2 using the Schur formula, and $\xi \equiv d^2$ will be

minimized, as the overall semidefinite programming setting is a particular case of convex optimization.

Results are checked numerically and plotted for three examples, including an example where the ellipses do intersect and, therefore, the distance is zero (although the numerical solver does not yield exactly zero as a result, due to the tolerances and finite precision of the code). The actual software used in this video is Matlab, jointly with the free SDP toolboxes YALMIP and SeDuMi.

[904: distlmiSetEN] Distance between 'LMI sets': semidefinite programming, intersection of ellipses example SeDuMi, Yalmip.

06:31

**** 

Materiales: [[Cód.:](#) DistanceLMIsetsENG.mlx] [[PDF](#)]

[[PLAY](#) ►]


This video generalises the ideas in video [903] to computing the distance between two “LMI sets”, i.e., sets defined by the points $x \in \mathbb{R}^n$ such that a given $LMI(x) \succeq 0$ holds. Ellipses are LMI sets, but so they are the intersection of ellipses, convex hull of ellipses, direct sum of ellipses, polyhedra and, in general, a class of convex shapes with polynomial boundaries (do search for works by D. Henrion from LAAS on geometry of LMI sets if you are interested in these topics).

The simplest case is computing the minimum distance between intersections of ellipsoids, so the code of the above-referred video is modified (just a couple of lines need modifications) to illustrate how LMI/SDP can handle the problem in a straightforward way (up to some numerical computation tolerances, of course).

31.2 Control de coste garantizado

El problema de control óptimo LQR resuelto mediante desigualdades matriciales lineales fue abordado en el vídeo [lqrlmi(09:35)], aunque no aportaba en el caso lineal nada nuevo respecto a la solución clásica basada en ecuación de Riccati. Aquí se presentan extensiones del mismo a problemas robustos, no lineales, etc. que ya sólo son posibles en el enfoque LMI.

[905: lqrlmiro] Control robusto (coste LQR garantizado) de procesos lineales politópicos con desigualdades matriciales lineales (LMI): teoría y ejemplo Matlab

****  11:00

Materiales: [[Cód.:](#) lqrlmitestRobust.mlx] [[PDF](#)]

[[YouTube](#) ►]

Este vídeo adapta la teoría del vídeo [595] al caso *robusto* (plantas lineales con incertidumbre o posible variación temporal en sus parámetros).

El caso que se considera es el de sistemas *politópicos*, expresables como una combinación convexa $\dot{x} = \sum_{i=1}^r h_i(t) \cdot (A_i x + B_i u)$ de modelos vértice $A_i x + B_i u$, cuyos coeficientes de interpolación h_i verifican $0 \leq h_i \leq 1$ y $\sum_{i=1}^r h_i = 1$.

El vídeo comienza revisando dichas nociones de sistemas politópicos, generalizando el caso de la “interpolación” entre los extremos de un intervalo.

Una vez introducida la clase de sistemas a tratar, el control robusto se propone como una realimentación del estado lineal $u = -Kx$, esto es, con idéntica estructura al del caso lineal LQR. Las LMIs resultantes son muy parecidas, la diferencia es que ahora deben verificarse simultáneamente para todos los r modelos vértices.


A partir del minuto [08:30] se plantea un ejemplo numérico de LMIs sobre un modelo politópico planteado en la parte inicial del vídeo.

Como material recomendable adicional, otro ejemplo de modelado politópico se discute en el vídeo [907], y el cálculo del controlador robusto para dicho modelo (con la misma teoría

que aquí, comparando con planificación de ganancia) se desarrolla en el vídeo [910]. La teoría de planificación de ganancia (cuando h_i es conocida) se aborda en los vídeos [906] y [909].

[906: lqrgs1] [Control con planificación de ganancia para sistemas politópicos via LMIs: planteamiento e introducción](#)

Materiales: [lqrlmiGainSchedulingTeoria.pdf]

****  11:58

[YouTube ▶]

Este vídeo plantea el problema de control óptimo (coste garantizado) mediante planificación de ganancia en sistemas politópicos $\dot{x} = \sum_{i=1}^r h_i(t)A_i x + B_i u$, con $0 \leq h_i \leq 1$, $\sum_{i=1}^r h_i = 1$. En los primeros 3 minutos, se introduce el problema y se diferencia del problema lineal $r = 1$, o del robusto $u(x)$, planteando la posibilidad de calcular un $u(h, x)$, que se denomina planificación de ganancia.

Los casos lineal y robusto han sido desarrollados en el vídeo [595] y [905], respectivamente; la visualización previa de estos dos vídeos es muy aconsejable, para comprender mejor las modificaciones que la planificación de ganancia requiere.

En concreto, en planificación de ganancia se propone $u = -(\sum_{i=1}^r h_i(t)F_i) \cdot x$.

A partir del minuto [04:30] se calculan las ecuaciones en bucle cerrado con este tipo de controladores, que resultan en sumas dobles $\dot{x} = \sum_{i=1}^r \sum_{j=1}^r h_i(t)h_j(t) \cdot (A_i - B_i F_j)x$.


Por conveniencia, se agruparán los términos de esa doble suma de cierta forma que luego será útil en las condiciones LMI: la suma convexa será negativa si todos los sumandos lo son.

La parte final del vídeo usa estas expresiones del bucle cerrado y las agrupaciones para plantear LMIs en una variable P que acota el coste en función de las condiciones iniciales con $x_0^T P x_0$ si el conjunto de ganancias K_j está calculado a priori, en lo que se conoce como condiciones LMI de *análisis*. Estas condiciones de análisis son usadas en un ejemplo numérico en el vídeo [908] para comprobar si un cierto diseño podría funcionar o no.

Nota: las condiciones de análisis presentadas son muy poco útiles, dado que ideas intuitivas como un diseño "separado" de K_i en función de A_i y B_i son incorrectas, existiendo muchísimos contraejemplos en la literatura donde controladores "individuales" óptimos son inestables al ser combinados; las ideas que explican los errores inherentes a este enfoque son ilustradas, mediante un ejemplo numérico, en el vídeo [908]. Por ello, se enuncian simplemente a título pedagógico, para introducir conceptos que serán explotados y mejorados en las condiciones de síntesis que se discuten en el vídeo [909], continuación y mejora de éste. Las condiciones de análisis aquí presentadas, pues, serán usadas con muy poca frecuencia.

[907: modelgs] [Modelado de sistemas lineales politópicos para planificación de ganancia](#)

Materiales: [[Cód.:](#) ModelPolitopicParaPlanifGan.mlx] [[PDF](#)]


****  08:11

[YouTube ▶]

En este vídeo se discute el modelado de sistemas lineales con parámetros inciertos (LPV) $\dot{x} = A(\theta)x + B(\theta)u$ (con θ variantes en el tiempo o no-linealidades) en forma politópica $\sum_{i=1}^r h_i(x, t) \cdot (A_i x + B_i u)$, en base a un ejemplo; la idea es similar a la planteada en el ejemplo del vídeo [905], cuya visualización podría ser aconsejable. Se discute en concreto cómo calcular los h_i a partir de las medidas de θ , basándose en fórmulas de interpolación; este cálculo permite plantear reguladores por *planificación de ganancia*, según se plantea en el vídeo [906]; la metodología se completa en el vídeo [909].

[908: lqrgsno] [Planificación de ganancia interpolando diseños LQR lineales \(método ERRÓNEO, como prueban las LMIs\)](#)

Materiales: [[Cód.:](#) lqrlmiGainSchedulingEjemplos.mlx] [[PDF](#)]

***  10:59

[YouTube ▶]


Este vídeo plantea una posible solución (incorrecta) al problema de control de coste garantizado con planificación de ganancia en sistemas politópicos $\dot{x} = \sum_{i=1}^r h_i(t) \cdot (A_i x + B_i u)$ que peca de ingenua: diseñar un controlador LQR para cada vértice y “mezclarlos” como se mezcla el proceso, interpolándolos con las mismas h_i .

En efecto, el planteamiento es erróneo, porque el bucle cerrado es un polinomio cuadrático en las variables h , y el “peor caso” no tiene por qué darse en los vértices; en el ejemplo en concreto que se aborda (usando el modelo del vídeo [907]), efectivamente se encuentra un punto h intermedio donde los polos son casi tres veces más lentos, y el coste el doble (calculado con la función `lyap`), que en el peor de los ocho vértices. Aparte, la posibilidad de variación de los parámetros en el tiempo implica que el peor caso será, posiblemente, aún peor de lo calculado para ese h fijo.

Para comprobar de forma rigurosa la estabilidad y prestaciones de esta solución candidata se programan las LMIs de análisis (sólo P en la función de Lyapunov $x^T P x$ es variable de decisión, siendo las K_i de $u = -\sum_{i=1}^r h_i K_i x$ conocidas); el detalle de esas LMIs se analiza en el vídeo [906].

En este ejemplo numérico concreto, las LMIs salen **no factibles**; ello significa que no se puede garantizar estabilidad en bucle cerrado para variaciones temporales arbitrarias de los parámetros del modelo. Es posible que otras LMIs pudieran, o que se pudiese garantizar dicha estabilidad si se supusieran determinadas cotas en la velocidad de variación de los parámetros, pero ambas cuestiones están fuera de los objetivos de este material introductorio... además, la misma teoría de control óptimo con LMIs sí que permitirá obtener soluciones factibles para el mismo modelo, como se discute en el vídeo [910].

En resumen, el uso de técnicas de diseño de control *vértice a vértice* sin tener en cuenta la *interacción* se **desaconseja**, y este vídeo ilustra algunas de las razones por las que esta idea no debe ser usada... el planteamiento formalmente correcto es el del vídeo [909].


[909: lqrgs2] **Control con planificación de ganancia para sistemas politópicos:** *****  10:27
condiciones LMI de síntesis
Materiales: [lqrlmiGainSchedulingTeoria.pdf] [YouTube ▶]

Este vídeo es continuación del vídeo [906], donde se planteó el problema de planificación de ganancia en sistemas politópicos y se calcularon las ecuaciones de bucle cerrado, donde aparecían *dobles sumatorios* $\sum_{i=1}^r \sum_{j=1}^r h_i h_j \Theta_{ij}$. La visualización de este vídeo así como de los vídeos sobre el caso lineal [595] o robusto [905] se aconseja para poder comprender mejor las diferencias con el caso ahora considerado.

Las ideas básicas de ese vídeo anterior se revisan rápidamente en los primeros dos minutos, y a partir del minuto [02:25] se desarrollan las LMIs necesarias para diseño de ganancias (porque las LMIs del vídeo [906] sólo podían ser resueltas de forma eficiente si la única variable de decisión era la cota P).

Para obtener LMIs de síntesis se necesita hacer el cambio de variable $x = P^{-1}\xi$, como en el caso robusto (vídeo [905]). También, debido a que aparecen productos $F_i^T R F_j$, debe acotarse $F_i^T R F_j + F_j^T R F_i \preceq F_i^T R F_i + F_j^T R F_j$, donde una desigualdad $A \preceq B$ se entiende como $A - B \preceq 0$. Se demuestra en el vídeo dicha desigualdad matricial.

Los desarrollos continúan haciendo relajaciones de la suma doble, como en el vídeo [906], y haciendo complemento de Schur para eliminar los productos de variables F . Con ello resultan un conjunto de LMIs cuya solución, computacionalmente eficiente, permite calcular las ganancias de $u = -\sum_{i=1}^r h_i(t) F_i x$ y la cota de coste $x_0^T P x_0$.

[910: lqrgsm1] **Control robusto y por planificación de ganancia en un sistema** *****  10:59
de 8 vértices: ejemplo Matlab
Materiales: [CÓD.: lqrlmiGainSchedulingEjemplos.mlx] [PDF] [YouTube ▶]

Este vídeo compara las prestaciones alcanzables en control de coste garantizado de un sistema politópico (por realimentación del estado) en los casos:

1. Robusto (variables h desconocidas), $u = -Kx$; ver detalle de la teoría en vídeo [905].
2. Planificación de ganancia (h conocido), $u = -\sum_{i=1}^r h_i K_i x$; detalle de la teoría en vídeos [906] y [909].

El modelo utilizado es el planteado en el vídeo [907], que ya se usó (sólo análisis) en el vídeo [908].

Resolviendo las LMIs, se comprueba que las prestaciones del diseño por planificación de ganancia son mejores (cota de coste más reducida) que en el caso robusto. Sin embargo, si se elimina la dependencia de uno de los parámetros inciertos, entonces las LMIs robustas *vs.* planif. de ganancia producen los mismos resultados. En definitiva, la mejora de resultados conseguibles con planificación de ganancia respecto al caso *robusto* depende de cada problema.

El vídeo finaliza con una discusión y conclusiones respecto a las ventajas e inconvenientes de esta metodología.

Apéndice

Appendix A

Matemáticas para control multivariable

A.1 Matrices, geometría

[911: matrix] Matrices, inversas, espacio nulo, pseudoinversas
Materiales: [mgo.pdf]

**  09:38
[YouTube ▶]

En este vídeo se revisan muy rápidamente diferentes conceptos de álgebra lineal, que serán importantes en control e identificación de sistemas lineales:

- Conceptos clave de matrices: matriz inversa, transpuesta, simétrica.
- Matriz inversa: solución única de sistemas de ecuaciones.
- Matriz singular: espacio nulo, solución general de un sistema de ecuaciones lineales.
- Rango de una matriz.
- Pseudoinversa izquierda y derecha, definición.
- Ejemplo numérico con Matlab (`inv` , `null` , `pinv`).

[912: matrixG] Geometría y matrices: autovalores, autovectores, matrices
definidas positivas, normas, elipsoides
Materiales: [mgo.pdf]

***  10:16
[YouTube ▶]

Este vídeo revisa conceptos geométricos asociados a matrices:


- autovectores y autovalores (también llamados vectores y valores *proprios*).
- matrices (simétricas) definidas positivas.
- Producto escalar de vectores. Norma de un vector. Vectores ortogonales. Bases ortonormales.
- Diagonalización de matrices simétricas: vectores propios ortogonales.
- Esferas, elipsoides, raíz cuadrada de una matriz \approx factorización de Choleski.
- Ejemplo Matlab (`eig` , `norm` , `ezcontour`).

[913: poliedr] Poliedros: definiciones básicas
Materiales: [Poliedros.pdf]

***  05:14
[YouTube ▶]

En este vídeo, de forma muy introductoria, se describe brevemente un poliedro como un conjunto de desigualdades (intersección de hiperplanos), y los conceptos de “representación por caras” o “representación por vértices” del mismo.

[914: mapr] **Matrices de proyección**

***  10:23

Materiales: [matrproy.pdf]

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo discute el concepto de proyección ortogonal y oblicua, con un ejemplo gráfico 2D/3D y generaliza la idea a un espacio vectorial de dimensión arbitraria (finita), donde un operador lineal puede ser representado por una matriz, y dicha matriz es una proyección si $P^2 = P$. La proyección es ortogonal si la matriz asociada verifica $P^T(I - P) = 0$. Se discute su interpretación, sus autovalores (sólo pueden ser cero o 1), la relación entre ortogonalidad y simetría, y se ponen un par de ejemplos: la proyección ortogonal más utilizada se obtiene a partir de la pseudoinversa de una matriz; también se construye (sin demostración) la proyección oblicua sobre el espacio columna de A en la dirección B .

[915: derivs] **Derivadas parciales, totales y jacobianos en funciones de varias variables**

**  10:13

Materiales: [Derivadas.pdf]

[YouTube ▶]

En este vídeo se discute el concepto de derivada parcial de una función de varias variables a los números reales y su ordenación formando un vector fila de derivadas parciales.

Para funciones $f : \mathbb{R}^n \mapsto \mathbb{R}^m$, se define la matriz jacobiana de derivadas parciales.

Se enuncia la regla de la cadena en el caso real y en el caso multivariable (multiplicación de jacobianos).

En el caso de tener una función $f(r, t)$ donde a su vez r sea función del tiempo, la derivada total es la derivada de $f(r(t), t)$ respecto al tiempo. La regla de la cadena permite calcularla en función de las derivadas parciales de f y de las velocidades dr/dt .

Ejemplos Matlab (Symbolic toolbox) aparecen en el vídeo [916]. Una aplicación de las ideas a un problema mecánico (cinemática) aparece en el vídeo [60].

[916: derivsml] **Derivadas parciales, totales y jacobianos: ejemplo Matlab (symbolic toolbox)**

***  10:35


Materiales: [[CÓD.:](#) DerivadasML.mlx] [[PDF](#)]

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este video presenta ejemplos Matlab de las ideas sobre derivadas parciales, jacobianos, regla de la cadena y derivadas totales presentadas en el vídeo [915]. Usa los comandos `syms`, `diff`, `jacobian` y `subs` para ilustrar los diferentes conceptos. Una aplicación a un problema mecánico (cinemática) aparece en el vídeo [60].

[917: vari] **Variedades diferenciables: representación implícita/explicita**

****  12:14


Materiales: [GeomDif1.pdf]

[PLAY ▶]

En este vídeo se describe brevemente el concepto de variedad diferenciable como el conjunto $S = \{x : f(x) = 0\}$ o el conjunto $S = \{x : \exists q \ x = \phi(q)\}$; el concepto de variedad (*manifold*, en inglés) generaliza el concepto de curva o superficie. La primera expresión se denomina representación *implícita* y la segunda *paramétrica*. Se pone a una esfera como ejemplo.

Existen determinadas condiciones de rango de los jacobianos de f y ϕ (vídeos [915], [916]) para que estén bien definidas las variedades; también dichos jacobianos verifican una condición de ortogonalidad que se discute en la parte final del vídeo.

Cuando el jacobiano de ϕ pierde rango se llega a una singularidad, que es un concepto importante en la especificación de los espacios de trabajo factibles en robots, por ejemplo. Se pone como ejemplo la singularidad de la representación mediante dos ángulos vista al principio del vídeo en los “polos” norte y sur.

[918: varimec]	Variedades diferenciables: ejemplos de aplicación a Mecánica (cinemática)	****  14:30
Materiales: [GeomDifEnMecanica2.pdf]		[PLAY ►]


Este vídeo desarrolla ejemplos de representaciones implícita y paramétrica de variedades diferenciables (vídeo [917]) aplicados a mecánica.

Los dos primeros minutos resumen rápidamente las definiciones fundamentales del vídeo [917], y luego se pone como ejemplo de variedad diferenciable S el lugar geométrico de las posiciones de un péndulo esférico (2 grados de libertad): una variedad 2-dimensional en \mathbb{R}^3 .

El lugar geométrico de las velocidades admisibles para un cierto punto p se denomina *espacio tangente* en p , $T_p S$; se obtiene mediante jacobianos de las restricciones de posición. Es calculado en el ejemplo del péndulo y también se recuerda la relación de $T_p S$ con la linealización de funciones de varias variables (vídeo [83]).

Como segundo ejemplo se pone el lugar geométrico de posiciones y velocidades de un sistema formado por dos masas puntuales unidas por una barra de masa e inercia despreciable: una variedad 5-dimensional en \mathbb{R}^6 . El lugar de las velocidades es, también, 5-dimensional. El espacio de estados posición-velocidad compatible con unas restricciones de posición (en mecánica, se denominan restricciones *holonómicas*) es un conjunto 10-dimensional que se denomina en geometría diferencial como *fibrado tangente*.

A.1.1 Elipses y Elipsoides

[919: ellip1]	Elipsoides, matrices definidas positivas (1): definiciones básicas y motivación	**  15:56
Materiales: [ElipsIntroES.zip]		[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo presenta motivación y definiciones básicas de: esfera centrada en el origen $x^T x \leq 1$, elipsoide alineado con ejes coordenados $x^T D x \leq 1$ (D matriz diagonal) y elipsoide genérico $x^T P x \leq 1$ (P matriz simétrica definida positiva), introduciendo matrices de rotación (matrices ortogonales) y la diagonalización para explicar las razones que llevan a esa última definición (la idea se discute en más profundidad en el vídeo [920]).

Presenta ejemplos Matlab con `fimplicit`, y motiva su uso en mínimos cuadrados (geometría de esferas), mínimos cuadrados ponderados (geometría de elipses) y otros ámbitos de control, robótica, estadística, materiales, etc.

El vídeo [920], continuación de este, discute en detalle la diagonalización, los elipsoides no centrados en el origen y tres formas equivalentes de representarlos.

[920: ellip2] **Elipsoides (2): interpretación geométrica diagonalización, formas alternativas expresión elipsoides**

Materiales: [ElipsIntroES.zip]

** 13:46

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo, continuación del [919], insiste en el significado de la diagonalización de una matriz positiva definida, para determinar semiejes y sus direcciones (recuperando los escalados diagonales y rotaciones que lo generan).

La segunda parte del vídeo discute formas alternativas de representar elipsoides: como $x^T P x \leq 1$, como $x^T Q^{-1} x \leq 1$ o como la imagen de la esfera unidad bajo una transformación lineal representada por una matriz L . También presenta el elipsoide no centrado en el origen.

El vídeo [921], continuación de éste, presenta ejemplos 3D y casos degenerados de P o Q no invertibles.

[921: ellip3] **Elipsoides (3): ejemplo 3D, casos degenerados, hiperboloides**

Materiales: [ElipsIntroES.zip]

** 09:38

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo presenta ejemplos de elipsoides en 3D para completar los ejemplos de los vídeos [919] y [920].

También presenta casos degenerados cuando P o Q no son matrices invertibles en expresiones $x^T P x$ o $x^T Q^{-1} x$. En ese caso (si los autovalores no nulos son positivos) se da lugar a cilindros o a elipsoides de menor dimensión.

La última parte del vídeo discute el caso donde P tenga valores propios negativos, que generan hiperboloides.

[922: ellip4] **Elipsoides (4): equivalencia representaciones**

Materiales: [ElipsFull.zip]

**** 13:47

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo presenta una discusión sobre equivalencia de representaciones de elipsoides: (1) directa $x^T P x \leq 1$, (2) inversa $x^T Q^{-1} x \leq 1$ y (3) como transformación lineal de una esfera $x = Lu$, $u^T u \leq 1$.

Este vídeo podría considerarse opcional si estás comenzando con todo esto, dado que discute refinamientos que podrían no ser necesarios para un primer acercamiento a estos temas.

Los temas aquí discutidos completan las ideas presentadas en el vídeo [920], anterior a éste, en concreto cómo transformar la representación (3) a la representación (2) cuando L no es invertible, bien por tener exceso de columnas o exceso de filas.

La parte final del vídeo discute la transformación de la representación (2) a la (3) mediante la factorización $Q = LL^T$ bien por método de Choleski (L triangular) o mediante diagonalización ($L = V\sqrt{D}V^T$ simétrica a partir de la diagonalización de $Q = VDV^T$). Esta L no es única porque queda un grado de libertad de “rotación” dado que la esfera $u^T u \leq 1$ permanece invariante cuando es rotada, de modo que existen una infinidad de L que representan a la misma esfera o elipsoide (la raíz cuadrada de una matriz no es única).

[923: ellip5] **Elipsoides (5): inclusión**

Materiales: [ElipsFull.zip]

**** 12:13


[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo desarrolla las condiciones para que un elipsoide esté incluido en otro. Básicamente, en forma directa, $x^T P_1 x \leq 1$ está incluido en $x^T P_2 x \leq 1$ si y sólo si $P_1 - P_2 \succeq 0$. Si se expresan en forma inversa, $x^T Q_1^{-1} x \leq 1$ está incluido en $x^T Q_2^{-1} x \leq 1$ si y sólo si $Q_2 - Q_1 \succeq 0$.

El vídeo dedica dos terceras partes de su duración a la demostración de las condiciones anteriores. Como caso particular, si uno de los dos elipsoides es una esfera de radio ρ , esto es $x^T \rho^{-2} x \leq 1$, la aplicación de las anteriores condiciones resulta en condiciones sobre los valores propios de P o Q que dan el radio mínimo y radio máximo del elipsoide (radio de las esferas inscritas y circunscritas, respectivamente).

[924: ellip6] Elipsoides (6): cortes, transformaciones, proyecciones

***  08:48

Materiales: [ElipsFull.zip]

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo, parte de la serie que explora las propiedades de los elipsoides (vídeos [919], [920] y sucesivos), discute el corte de un elipsoide con un plano o subespacio, la transformación lineal de un elipsoide y, como casos particulares, los cortes con los planos coordenados ($x_i = 0$), y la proyección sobre los mismos.

[925: ellip7EN] Ellipsoids (7): relationship with singular value decomposition (SVD)

****  09:41

Materiales: [ElipsFull.zip]

[YouTube ►]

* [ENLACE A SPANISH VERSION](#)

This video discusses the relationship between the singular value decomposition $L = USV^T$ of a matrix L with the geometry of the linear transformation with said matrix L of a unit sphere $x = L\nu$, $\|\nu\|_2 \leq 1$.

It is shown that the directions of the semiaxes are the columns of U , that the lengths of the semiaxes are the singular values (diagonal of S), and that the columns of V (input directions) are the points of the unit sphere that multiplied by L become the endpoints of the semiaxes of the ellipsoid.

A.2 Optimización

[926: optimi] Optimización sin restricciones y con restricciones de igualdad.
Revisión teoría y ejemplo Matlab

**  06:44

Materiales: [mgo.pdf]

[YouTube ►]


Este vídeo revisa las ideas básicas de optimización que se ven en los primeros cursos de carreras técnicas:

- Optimización: definición de función de coste.
- jacobiano (primeras derivadas) y hessiano (segundas derivadas) de una función.
- Mínimos locales en funciones diferenciables: jacobiano nulo es condición necesaria.
- Utilidad de la matriz hessiana: positiva definida implica mínimo, negativa definida implica máximo, indefinida implica punto de silla.
- Optimización sujeta a restricciones de igualdad: multiplicadores de Lagrange.
- Ejemplo numérico de optimización con Matlab (`diff`, `jacobian`, `solve`, `eig`).

[927: optimfmga]

Optimización local (fmincon) versus global (algoritmo genético): ejemplo Matlab

Materiales: [Cód.: minimlocalvsglobal.mlx] [PDF]

**  05:56

[YouTube ▶]

En este vídeo se compara la solución de código de optimización local (`fminunc` de la *Optimization Toolbox* con la obtenida por `ga` de la *Global Optimization Toolbox*. La optimización local es rápida pero se puede quedar atrapada en mínimos locales, mientras que la optimización por algoritmo genético (búsqueda aleatoria dirigida) la probabilidad de obtener el mínimo global es alta a cambio de un coste computacional bastante mayor. La parte final del vídeo propone, como opción intermedia, el hacer varias optimizaciones locales con condiciones iniciales seleccionadas al azar en una región de búsqueda.


Un ejemplo de uso de algoritmos genéticos en un problema de identificación experimental de un modelo masa-muelle-amortiguador aparece en el vídeo [749]; allí los métodos locales no funcionaron del todo bien, por eso se decidió hacer la grabación sólo con `ga`.

A.2.1 Mínimos cuadrados

[928: optimLS]

Optimización de funciones cuadráticas. Mínimos cuadrados. Revisión teoría y ejemplo Matlab.

Materiales: [mgo.pdf]

**  06:04

[YouTube ▶]

En este vídeo se revisa la optimización de una función cuadrática $J(x) = m + r^T x + \frac{1}{2} x^T Q x$. Su jacobiano es $r^T + x^T Q$, y su hessiano es Q . Con ello se puede demostrar que si Q es definida positiva, se alcanza un único mínimo en $-Q^{-1}r$.

A continuación, sobre un sistema de ecuaciones $Ax = b$, se enuncian el problema de mínimos cuadrados con más ecuaciones que incógnitas (minimizar $\|b - Ax\|^2$) y el otro problema de mínimos cuadrados con más incógnitas que ecuaciones (minimizar $\|x\|^2$ sujeto a $Ax - b = 0$). Expresando el coste como $J(x)$ arriba (usando multiplicador de Lagrange en el segundo caso) y calculando los adecuados jacobianos y hessianos, se demuestra que las soluciones de estos problemas de mínimos cuadrados vienen dadas en función de la pseudoinversa izquierda de A y de la pseudoinversa derecha, respectivamente.

El vídeo finaliza con un sencillo ejemplo numérico con Matlab y `pinv` de mínimos cuadrados.


La generalización al caso de mínimos cuadrados *ponderados* y su utilidad en problemas de control (conseguir $y = r$ con un modelo $y = Gu$) se abordan en los vídeos [929] (más filas que columnas) y [930] (más columnas que filas).

Nota: Los mínimos cuadrados considerados en este vídeo tienen, además, una interpretación geométrica (proyección ortogonal) y estadística (mínima varianza). Estas cuestiones se discuten en el vídeo [971].

[929: mcpy]

Mínimos cuadrados ponderados en control (I): exceso de salidas

Materiales: [Cód.: MinCuadPondContrVC.mlx] [PDF]

**  13:09

[YouTube ▶]

Este vídeo discute la aplicación a un problema de control (estático) $y = Gu$ con 12 salidas y 4 entradas de la pseudoinversa para ajustar u por mínimos cuadrados para intentar conseguir minimizar $\|r - Gu\|_2$. La pseudoinversa se discute en la revisión matemática sobre matrices (vídeo [911]) y en la solución del problema de mínimos cuadrados (vídeo [928]).

Tras plantear los mínimos cuadrados no ponderados $u^* = \text{pinv}(G)r$, se plantea un caso ponderado donde se construye un peso para minimizar $\sum_{i=1}^1 2(w_i(r_i - y_i))^2$.

La parte final del vídeo plantea construir el peso como la inversa de unos “umbrales deseados de error”, por ejemplo se desea que $|e_i| < \beta_i$ siendo β_i arbitrario. Se construye un error escalado $e_{esc,i} = e_i/\beta_i$ y se plantea un problema de mínimos cuadrados ponderados asociado, de modo que si el error escalado es menor que 1 (bien como máximo de valor absoluto de elementos –geometría de hipercubos–, bien como norma del vector –geometría de esferas–) se garantiza que los errores están por debajo del umbral buscado.

Aunque formalmente es equivalente al “peso w_i ” estos umbrales y escalados facilitan cálculos e interpretaciones prácticas, por ejemplo en los escalados de las salidas para controlabilidad entrada-salida del vídeo [433].

[930: mcpu] **Mínimos cuadrados ponderados en control (II): exceso de entradas**

Materiales: [[CÓD.:](#) MinCuadPondContrVM.mlx] [[PDF](#)]

** 09:20

[YouTube ▶]

Este vídeo discute la aplicación a un problema de control (estático) $y = Gu$ con 2 salidas y 4 entradas de la pseudoinversa para ajustar u por mínimos cuadrados para intentar conseguir minimizar $\|u\|_2$ entre todas las soluciones tales que $r = Gu$, siendo r una referencia deseada para la salida y . La pseudoinversa se discute en la revisión matemática sobre matrices (vídeo [911]) y en la solución del problema de mínimos cuadrados (vídeo [928]).

Tras plantear los mínimos cuadrados no ponderados $u^* = \text{pinv}(G)r$, se plantea un caso ponderado donde se construye un peso para minimizar $\sum_{i=1}^4 (w_i u_i)^2$ sujeto a $r = Gu$.

La parte final del vídeo plantea construir el peso como la inversa de unos “umbrales disponibles para cada actuador”; por ejemplo, debido a saturación, se desea que $|u_i| < \gamma_i$ siendo γ_i determinado por el incremento disponible hasta saturar. Se construye una entrada escalada $u_{esc,i} = u_i/\gamma_i$ y se plantea un problema de mínimos cuadrados ponderados asociado, de modo que si el control escalado es menor que 1 (bien como máximo de valor absoluto de elementos –geometría de hipercubos–, bien como norma del vector –geometría de esferas–) se garantiza que los actuadores están por debajo del umbral buscado.

Aunque formalmente es equivalente al “peso w_i ” estos umbrales y escalados facilitan cálculos e interpretaciones prácticas, por ejemplo en los escalados de las entradas para controlabilidad entrada-salida del vídeo [433].

*Puedes ver un caso de estudio de la interpretación de estos mínimos cuadrados en control en el vídeo 584, en la página 277.

A.3 Variable Compleja

- ▶ *Repaso de variable compleja:* [14:30]
[Enlace]
- ▶ *Relación de Euler y potencias de un complejo:* [Andrés Camacho]
[Enlace]

A.4 Descomposición en valores singulares (SVD)

[931: svdOA] **Descomposición en valores singulares (SVD): introducción rápida**

Materiales: [SVDbreve.pdf]


*** 11:35

[YouTube ▶]

Este vídeo presenta una introducción rápida al concepto e interpretación de la descomposición en valores singulares (sin demostración, discutida en el video [932]).

- Enunciado del teorema de descomposición $A = USV^T$.
- Esbozo de cómo obtener U , S , V por diagonalización de $A^T A$ o AA^T .
- Interpretación geométrica: proyección \rightarrow escalado diagonal \rightarrow rotación.
- Ganancia máxima de una matriz.
- Rango “práctico” de una matriz proveniente de datos experimentales (imprecisos, con ruido): asumir valores singulares “comparativamente pequeños” como cero.
- Relación de la descomposición en valores singulares con inversas y pseudoinversas.

[932: svd1] Descomposición en Valores Singulares: teorema y demostración
Materiales: [SVDdemoOA.pdf]

*****  10:26
[YouTube ►]

Este vídeo presenta formalmente el teorema de descomposición $A = USV^T$ de una matriz (U ; V ortogonales, S diagonal de valores singulares) y procede a su demostración. Por tanto, establece con más precisión algunos de los conceptos esbozados en los primeros minutos del video [931].

El video enuncia el teorema de descomposición en valores singulares de una matriz real, en versión “económica” y desarrolla su demostración. Al final, como corolario, se propone la descomposición “completa”.

De todas formas, conocer el detalle de la demostración no es necesario para comprender la interpretación geométrica presentada, por ejemplo, en el video [931]. El visionado del presente vídeo podría considerarse opcional, y está indicado una vez visionado el resto de los vídeos sobre este tema, si se está interesado en el detalle de la demostración para una mejor comprensión del SVD en su conjunto. La relación con la geometría de elipsoides se aborda en el vídeo [??].

[933: svd2] Descomposición en valores singulares (SVD): propiedades
Materiales: [SVDf.pdf]

*****  17:10
[YouTube ►]

Este vídeo, continuación del [931], discute la interpretación y utilidad de la descomposición en valores singulares en algunos casos.

- Revisión de interpretación geométrica, y de relación con inversas y pseudoinversas. El detalle de la interpretación geométrica y su relación con la geometría de las elipses y elipsoides se aborda en el vídeo [??] con mayor detalle.
- Relación de la descomposición con la computación del espacio nulo de una matriz.
- Rango como número de valores singulares no nulos, rango “práctico” aproximado.
- Detalle de concepto de “ganancia” de una matriz: ganancia “escalar” 1×1 , ganancia direccional, caso general ganancia de una matriz.
- Ganancias máxima y mínima. Demostración de su obtención mediante multiplicadores de Lagrange.
- Propiedades de las ganancias máxima y mínima.
- Conclusiones e introducción a disciplinas donde resulta útil la descomposición.

[934: gevd] Valores propios generalizados $[V,D]=\text{eig}(A,B)$

***** 
Materiales: [gevp.pdf]


Este material discute (muy brevemente) el concepto de valor propio o autovalor generalizado $\text{eig}(A, B)$, planteando propiedades de interés en dinámica de sistemas y estadística. No se discuten las consecuencias e interpretación cuando B no es invertible, que es la verdadera motivación detrás de la definición, se remite al lector a textos de Matemáticas adecuados para ello.

Appendix B

Introducción a Estadística para ingenieros de control y procesos

Nota: El objetivo de estos materiales es dar un “resumen rápido” de las ideas principales de interés para un ingeniero que desee iniciarse al trabajo con procesos multivariados sujetos a ruido. Algunos resultados son intencionalmente inexactos, para más simplicidad y brevedad –usualmente asumiendo infinitas muestras y distribución normal–, y otros son inexactos por omisiones y detalles que los verdaderamente expertos en Estadística conocen mucho mejor que el autor (mis disculpas por ello). Si el lector de estos materiales desea especializarse en Estadística y Análisis Multivariante, se le remite a múltiples libros de texto con presentaciones más detalladas y rigurosas de los conceptos que aquí se esbozan.

B.1 Motivación

[935: estic]	Relevancia de la estadística en ingeniería de control/procesos (generalidades)	**  10:41
Materiales: [statisticsincontrol.pdf]		[YouTube ►]

Este vídeo discute las razones por las que la estadística es una disciplina cuyo estudio resulta importante en ingeniería de control, optimización de procesos industriales, etc. y en general, para expertos en ingeniería de sistemas y automática.

Básicamente, la estadística es importante en estas facetas de la ingeniería porque el objetivo del “control” en términos “amplios” es reducir la “variabilidad” de determinadas señales del sistema (variables controladas) y la estadística es la especialidad de las Matemáticas que estudia dicha variabilidad (ante ciertas suposiciones sobre su origen).

La variabilidad puede ser debida a que haya parámetros de modelos que no tomen los valores que se suponía en el momento de diseño, o bien por la aleatoriedad de las perturbaciones. La primera de las cuestiones obliga a hacer “identificación” basada en datos (que, obviamente, están contaminados por ruido de medida y ruido de proceso); la segunda de las cuestiones obliga a hacer control para “rechazo de perturbaciones”, idealmente, control de “mínima varianza”.

Nótese que los errores por dinámica no modelada o no-linealidades no tienen una interpretación “probabilística” y, por tanto, deben ser abordadas con técnicas de control *robusto*, discutidas en otros materiales de la colección.

B.2 Conceptos básicos: variables aleatorias, probabilidad, media, varianza

[936: vagd] Variables aleatorias: generalidades, caso discreto

Materiales: [vagd.pdf]

12:19

[YouTube ▶]

Este vídeo presenta las definiciones genéricas sobre variables aleatorias (resultados “inciertos” de un experimento) y su espacio probabilístico asociado: variable, eventos, medida de probabilidad, axiomas, tipos de variables aleatorias (categóricas *vs.* numéricas) y ejemplos.

[937: vard] Variables aleatorias reales: función de densidad, función de distribución acumulativa, vbles. Mixtas

Materiales: [vard.pdf]

13:34

[YouTube ▶]

Este vídeo detalla los conceptos asociados a la función de densidad de una variable real, dado que la probabilidad de un elemento numérico en una variable real es, en general, nula. La función de densidad puede ser entendida de manera “informal” como la probabilidad de un intervalo infinitesimal.

Dado que el evento más usual en las aplicaciones suele ser calcular la probabilidad de que una variable aleatoria esté por encima o por debajo de un umbral, se define la función de *distribución* acumulativa (cumulative distribution function, CDF, en inglés) como la integral de la densidad desde $-\infty$ hasta ese umbral que está siendo evaluado.

También se explica el concepto de variables “mixtas” con probabilidad distribuida con una densidad excepto en un conjunto de puntos aislados con probabilidad “concentrada”. Este sería el caso, por ejemplo, de un sensor sujeto a saturación. En estos casos la función de densidad puede incluir términos de integral finita en un intervalo infinitesimal (delta de Dirac).

A veces, una variable aleatoria se produce como resultado de transformar otra variable aleatoria pasándola a través de una función. La función de densidad resultante se aborda en el vídeo [941].

[938: mvmp] Parámetros (media, varianza, momentos) de una distribución de probabilidad

Materiales: [ParamsDistribPr.pdf]

11:05

[YouTube ▶]

En este vídeo se describen los parámetros que ayudan a comprender el significado de una distribución de probabilidad: la *media* (valor alrededor del cual están las distintas “realizaciones” –experimentos, muestras– de la variable aleatoria), la *varianza* (media del cuadrado de incrementos), y la *desviación típica*, que es una medida cualitativa de la dispersión esperada alrededor de la media.

La idea se generaliza a la esperanza matemática de cualquier función, y se definen los momentos de mayor grado (grado 3: sesgo, grado 4: curtosis) a partir de ella, indicándose brevemente su interpretación “informal” en términos de “asimetría” y “porcentaje de outliers”.

En muchos de los cálculos en este vídeo se requiere restar la media de una variable. Ello define lo que se denominarán variables *incrementales*. Eso también ocurrirá en muchos de los desarrollos posteriores. Estas variables incrementales alrededor de la media son el análogo a lo que en control denominábamos linealización y variables incrementales alrededor de un punto de funcionamiento.

Nota, en este vídeo se discute cómo obtener los parámetros a partir de un modelo de la función de densidad o probabilidad discreta. En el vídeo [939] se discutirá la relación de estos con los cálculos a partir de un conjunto de muestras.


[939: muestr] Estadísticos muestrales (media, varianza, ...)

***  12:58**Materiales:** [estadisticmuestral.pdf]

[YouTube ►]

Como se ha comentado anteriormente, las definiciones de media, varianza, y esperanza matemática requieren “formalmente” la ejecución de integrales (o sumas en variables discretas) sobre la densidad de probabilidad. Sin embargo, en muchas aplicaciones dicha función de densidad es parcialmente (o totalmente) desconocida, y de lo único que se dispone es de datos (muestras) recogidos en el pasado sobre la variable aleatoria bajo estudio.

En ese caso, se definen los estadísticos muestrales: media muestral $\frac{1}{N} \sum_{i=1}^N x_i$, varianza muestral, etc. y, en general, un “valor esperado muestral” promedio que sustituye a la esperanza matemática en términos formales, que aproximan a los parámetros “verdaderos”, pero siendo obtenidos a partir de datos experimentales.

Se comentan brevemente las leyes de los “grandes números” (que indican que los parámetros muestrales convergen a los verdaderos cuando el número de muestras N tiende a infinito) y se esboza la existencia de resultados (de demostración más compleja) que, con un número finito de muestras, garantizan estimados PAC (probablemente aproximadamente correctos) de los parámetros de la distribución, como, por ejemplo, poder afirmar que la probabilidad de que la media “real” μ (usualmente desconocida) de una distribución esté alejada más de 0.25 unidades de la media muestral es menor del 2%. **** 

B.3 Variables aleatorias: caso multivariable

B.3.1 Transformación (función) de una variable aleatoria

[940: opvau] Transformación de una variable aleatoria: ejemplo distrib. uniforme (motivación)

**  19:36**Materiales:** [CÓD.: FuncsVbIsAleatUniformeSISO.mlx] [PDF]

[YouTube ►]

En este vídeo se plantea el problema de tener determinada función $y = h(x)$ donde la entrada sea una variable aleatoria. Por tanto y será una variable aleatoria (si se mira “aislada”, porque si se mira junto a ‘ x ’ la relación es determinista).

En este vídeo, sólo para el caso de distribución uniforme, se calculan histogramas, medias, varianzas y desviación típica de x e y , de cuatro formas:

1. A partir de muestras
2. Como esperanza matemática de funciones de x , usando la función de densidad de x
3. Como esperanza matemática de y e $(y - \bar{y})^2$, usando la densidad (marginal) de y , sin tener en cuenta su “dependencia” de x .
4. A partir de la media y la varianza de x (sólo válido en el caso $h(x)$ lineal)


Con número de muestras tendiendo a infinito, las cuatro opciones dan el mismo resultado.

La parte final del vídeo plantea las dificultades que origina una transformación no lineal (deformación, incluida discontinuidades, de la función de densidad original).

Una forma más correcta de tratar las transformaciones (funciones) de variables aleatorias en otras es con la denominada *fórmula del cambio de variable*, que se detalla en el vídeo [941]; de hecho, al final de dicho vídeo se recupera el caso uniforme tratado aquí. Una vez ese desarrollo teórico se ha completado, el vídeo [942] hace un análisis parecido al aquí desarrollado, pero con distribución normal.

[941: opvafcv]

Transformación (función) de una variable aleatoria: fórmula del cambio de variable (distr. continua)

***  11:57

Materiales: [[CÓD.:](#) FuncsVbIsAleatNormalSISO.mlx] [[PDF](#)]

[YouTube ▶]

En este vídeo se plantea el problema de tener determinada función $y = h(x)$ donde la entrada sea una variable aleatoria. Por tanto y será una variable aleatoria (si se mira “aislada”, porque si se mira junto a ‘ x ’ la relación es determinista).


En el vídeo introductorio [940], motivador a éste problema, se analiza una distribución uniforme de una forma “heurística”. El objetivo del presente material es presentar una fórmula de cambio de variable que aplique incluso a transformaciones no lineales (monótonas).

Esa fórmula calcula la función de densidad de y (salida) a partir de la función de densidad de x (entrada), utilizando la derivada de h y su inversa. La fórmula, en concreto, es $\frac{f(h^{-1}(y))}{|h'(h^{-1}(y))|}$. En la conclusión del vídeo se analiza que el caso uniforme en el vídeo arriba referido es una conclusión sencillísima de esta fórmula.

Nota: la relación entre dos variables aleatorias no tiene por qué ser “determinista” (es un caso extremo, siendo el otro extremo la “independencia estadística”). El enfoque general de variables aleatorias bidimensionales y su correlación, etc. se aborda en los vídeos [945] y siguientes.

[942: opvan]

Transformación (función) de una variable aleatoria: fórmula del cambio de variable, ejemplo distribución normal

***  12:15

Materiales: [[CÓD.:](#) FuncsVbIsAleatNormalSISO.mlx] [[PDF](#)]

[YouTube ▶]

En este vídeo se plantea el problema de tener determinada función $y = h(x)$ donde la entrada sea una variable aleatoria. Por tanto y será una variable aleatoria, aunque si se mirase junto a ‘ x ’ la relación fuese determinista.

En el vídeo introductorio [940], motivador a éste problema, se analiza una distribución uniforme de una forma “heurística”. La teoría de la *fórmula del cambio de variable* se detalla en el vídeo [941], y el presente vídeo (que resume en el primer minuto dicho vídeo teórico) presenta un ejemplo de su aplicación con una entrada x de distribución *normal* (estándar, de media cero y varianza unidad).

Primero, se aborda el caso de una transformación lineal $y = h(x) = 3x + 2$. Se comprueba que la fórmula del cambio de variable resulta en una distribución normal de media 2 y desviación típica 3, varianza 9.

La parte final del vídeo utiliza la fórmula de cambio de variable ante la función $y = h(x) = x^2$. Como no es monótona, se debe hacer con cuidado, considerando solamente la parte de $x > 0$ y luego argumentando en base a la simetría tanto de la densidad de x como de $h(x)$.


El resultado es una función de densidad $f_y(\cdot)$ “famosa”, que es la base de desarrollos posteriores (sobre varianza muestral), y se denomina χ^2 (de 1 grado de libertad) definida, efectivamente, como la distribución de probabilidad del cuadrado de una distribución normal estándar. Esos desarrollos no son, obviamente, objetivos de este vídeo.

B.3.2 Variables aleatorias multidimensionales

Introducción y planteamiento de los problemas a resolver

[943: vamintu1]

Variables aleatorias multidimensionales: motivación intuitiva (1)

*  16:11

Materiales: [VANDMotiv.pdf]

[YouTube ▶]

El análisis estadístico de una variable da para poco: calculo media, varianza, histograma o aproximación a la tabla/densidad de probabilidad y asunto concluido.

Lo importante en ciencia es analizar varias cosas de resultado incierto para intentar encontrar relaciones entre ellas: que lo que se creía aleatorio tiene al menos parte de determinista.

Para ello, hay que introducir las variables aleatorias multidimensionales, bien como conjuntos de variables aleatorias, conjuntos indexados o vectores. Este vídeo discute desde un punto de vista informal e intuitivo el qué son esas variables aleatorias multidimensionales y cómo organizarlas. Los problemas a resolver se discutirán en el vídeo [944], también de forma intuitiva y con ejemplos resolubles con “sentido común”. Un análisis teórico detallado de los conceptos se aborda en otros vídeos de la colección.

[944: vamintu2]

Variables aleatorias multidimensionales: motivación intuitiva (2), problemas a resolver

** 18:08

Materiales:

[VANDMotiv.pdf]

[YouTube ▶]

Este vídeo continua la discusión “informal” sobre qué es y qué objetivos pretende el análisis “combinado” de varias variables aleatorias (variables aleatorias multidimensionales), iniciada en el vídeo [943].

Aquí se esbozan intuitivamente el significado del análisis “marginal”, del análisis “conjunto”, del análisis “condicional dos a dos” y se da una pequeña pincelada de los objetivos de un análisis “multivariante completo” de una serie de datos (que si tiene muchas variables aleatorias y muchas muestras, se le apellida “Big Data”). El detalle de conceptos de covarianza, correlación, probabilidad condicional, etc. ya no es objetivo de este vídeo, y se remite a otros vídeos de esta colección.

Análisis

[945: va2d]

Variables aleatorias bidimensionales: distrib. marginal, covarianza

*** 12:14

Materiales:

[vblesaleat2D.pdf]

[YouTube ▶]

En este vídeo se presentan las variables aleatorias bidimensionales $(x, y) \in \mathbb{X} \times \mathbb{Y} \subseteq \mathbb{R}^2$ como paso preliminar para introducir las variables multidimensionales que son el objetivo del llamado análisis estadístico multivariante. Podría interesarte visualizar previamente una discusión “intuitiva”, sin fórmulas, sobre qué es y qué problemas resuelve el análisis de “múltiples” variables aleatorias (vídeos [943] y [944]).

El caso más sencillo sería el que una variable fuera una función (determinista) de otra, como se analiza en los vídeos [940] o [941]. Pero entre una relación totalmente “determinista” y, en el otro extremo, lo que se denomina “independencia estadística”, hay toda una serie de casos intermedios, que necesitan de las definiciones en este vídeo.

Se definen y calculan las distribuciones *marginales*, esto es, considerar a cada elemento como una variable aleatoria aislada y no considerando la existencia de la otra. Por ejemplo, la densidad marginal de x , que denotamos $f_x(\cdot)$ es $f_x(x) := \int_{\mathbb{Y}} f(x, y) dy$.

Con ello, se obtienen media, varianza y desviación típica marginales.

Como novedad, se introducen medidas de la *interacción* entre variables: la *covarianza* (media del producto) y la *correlación* (covarianza adimensionalizada dividiendo por las desviaciones típicas, motivada porque pasar de voltios a milivoltios multiplica la covarianza por mil sin tener realmente ningún significado diferente, de ahí la necesidad de esa medida adimensional de la “relación” entre variables).

Se motiva que la existencia de correlación está detrás de la predicción estadística y la regresión lineal, aunque dichos conceptos son objeto de otros materiales.

[946: prcond] Probabilidad y densidad condicional

***  09:28**Materiales:** [probcondicional.pdf]

[YouTube ►]

En muchas aplicaciones, se dispone de un modelo (o una aproximación del mismo basada en datos) de la distribución de probabilidad conjunta entre dos variables; además, una de ellas x es “medible” mientras que la otra y no lo es (o no tan fácilmente), pero y es importante dado que las decisiones a tomar en la aplicación pueden depender del valor de y . Por tanto, es conveniente usar las medidas de x para inferir una “predicción” o “estimación” de y ; esta predicción es una fórmula que depende de x cuya validez está, claro, *condicionada* a que pueda disponerse de un valor de x medido para aplicarla.

Motivados por la “correlación” entre variables introducida en el vídeo [945] este vídeo define el concepto de *probabilidad condicional* (o *densidad condicional* en variables reales) y justifica que es el concepto formalmente “correcto” para entender cuál es la mejor predicción de una variable dada otra (la correlación sólo era una “idea preliminar” sobre la interacción entre variables, relacionada con la regresión lineal).

La definición es, de hecho, la probabilidad (o densidad) conjunta dividida por la marginal. Se pone un ejemplo discreto de una producción de balones de dos tamaños y dos colores, postulando en ese caso que la mejor predicción “condicional” es la *moda* (valor más probable) de la probabilidad condicional. También se pone un ejemplo continuo de cómo se calcularía todo eso con integrales de la densidad.

Otro ejemplo sencillo de tablas de probabilidad condicional se aborda en el caso de estudio ‘tigre oculto’ de los vídeos [947] y [948], que se recomienda visualizar.

Nota: De la definición de la probabilidad condicional se desprende que $p(y|x)p_x(x) = p(x|y)p_y(y)$ (fórmula de Bayes). En el problema de estimar y a partir de x , la probabilidad marginal $p_y(\cdot)$ se suele denominar como probabilidad *a priori*, y la probabilidad condicional (una vez se dispone de la medida de x), se suele denominar como probabilidad *a posteriori*.

[947: tiger1] Tigre oculto (1): Probabilidad condicional, conjunta y a priori; planteamiento del problema

*  20:57**Materiales:** [[CÓD.:](#) Tiger1ES.mlx] [[PDF](#)]

[YouTube ►]

* [LINK TO ENGLISH VERSION](#)

Este vídeo es el primero de un caso de estudio de un problema “tigre oculto” que permitirá revisar y mejorar la comprensión de muchos conceptos de Estadística y probabilidad.

En este primer vídeo, se describe el problema (tigre que se encuentra en una de dos jaulas izquierda o derecha, pero cuyos rugidos a veces se escuchan por el lado equivocado porque el tigre ruge hacia un pasillo que conecta las puertas). El objetivo final del problema es averiguar dónde está el tigre (para “salvar la vida” saliendo por la otra puerta)... pero para eso hay que comprender muchos conceptos previos... empezando por las “tablas de probabilidad” que vemos aquí.

Aquí explicamos tablas de probabilidad condicional, de “observaciones” (escuchar rugidos izquierda o derecha) condicionadas a “localización” del tigre (tigre izquierda o derecha). Más definiciones y ejemplos del concepto de probabilidad condicional aparecen en el vídeo [946].

Tras ello, definimos el concepto de probabilidad “a priori” de que el tigre se encuentre en un lado u otro *antes* de escuchar el primer rugido, y con ello calculamos la probabilidad conjunta $p(m \wedge t) = p(m|t) \cdot p(t)$ de las cuatro combinaciones “atómicas” de “escuchar” y “localización tigre” posibles.

El vídeo discute la interpretación y relaciones entre esas tablas de probabilidad. En concreto, enfatiza que aunque a partir de la tabla “conjunta” se puede calcular todo (se

detalla en el vídeo [948], continuación de éste), resulta que dividir el problema en subproblemas resulta conceptualmente más sencillo con el concepto de probabilidad condicional (modelando el problema con un conjunto de afirmaciones del tipo “si se cumple cierta cosa entonces pasa esta otra cosa con tal o cual probabilidad”); de hecho, esta idea de “encadenar” tablas de probabilidad condicional da lugar a las “redes bayesianas”.

[948: tiger2] Tigre oculto (2): Probabilidad condicional y marginal a partir de conjunta; condicional directa/inversa

Materiales: [[CÓD.:](#) Tiger1ES.mlx] [[PDF](#)]

* 15:52

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo es el segundo del caso de estudio “tigre oculto” que comenzó en el vídeo [947] y que aquí se revisa brevemente.

En el citado vídeo se planteó el problema, se definió el modelo probabilístico a partir de una probabilidad de tigre izquierda/derecha al cerrar cierta puerta y una probabilidad condicional de por qué lado (izquierdo/derecho) se le oye rugir en función de (condicionado a) qué lado se quedó el tigre encerrado. Con ello se obtuvo la tabla de probabilidad conjunta.

En este vídeo, a partir de la probabilidad conjunta se obtienen las siguientes tablas de probabilidades:

- Marginal de localización de tigre (recupera la probabilidad de partida)
- Marginal de lado donde se escucha rugido
- Condicional de localización de tigre (condición) a lado de escucha de rugido (variable condicionada); esta ya la conocíamos, estamos desandando lo andado.
- Condicional de lado de escucha de rugido (condición) a localización de tigre (variable condicionada)

Más definiciones y ejemplos del concepto de probabilidad condicional aparecen en el vídeo [946].

La primera marginal y condicional tienen el significado discutido en el primer vídeo; la segunda marginal dice la frecuencia de rugidos izquierda o derecha “acumulada” si se repitiera muchas veces el experimento.

La segunda tabla condicional tiene un significado diferente, digamos que la cadena de causalidad va “al revés”: dado un efecto observado (rugido), se infiere la probabilidad “a posteriori” de sus dos posibles causas (tigre a la izquierda o tigre a la derecha). Esta idea es la base de la *inferencia estadística bayesiana*. El siguiente vídeo detallará más esta última idea.

[949: corrsg] Correlación estadística entre 2 variables: significado gráfico/geométrico

Materiales: [signifcorrelacion.pdf]


*** 10:40

[YouTube ▶]

Este vídeo discute el concepto de “correlación”, en concreto su interpretación “gráfica” (las variables con correlación elevada se pegan a una “línea”), y su interpretación geométrica: la varianza es una suma de cuadrados que, salvo escalado dividiendo por número de muestras, se parece a la norma euclídea (al cuadrado) y la covarianza se parece (salvo escalado) al producto escalar. La correlación, por tanto, se parece al coseno del ángulo entre dos vectores. Cuando la correlación es igual a +1 o a -1, esta interpretación geométrica indica que las muestras de una variables x son proporcionales a las muestras de y (determinismo perfecto). Por otro lado, la ausencia de correlación estadística tiene una interpretación geométrica de “ortogonalidad” entre muestras.

[950: indep1]

Independencia estadística: definición e ideas básicas sobre su significado


10:39

Materiales: [independenciaestad.pdf]

[YouTube ▶]


En este vídeo se define el concepto de “independencia estadística” entre dos variables aleatorias. La definición explicita que independencia se determina porque la densidad conjunta es el producto de densidades marginales. Equivalentemente, la condicional de una variable y dada otra x es igual a su marginal: el conocimiento de la otra variable no cambia nuestro estimado de la probabilidad de y respecto al que teníamos antes de asumir la existencia (y la disponibilidad de medidas) de x .

Dos variables estadísticamente independientes NO sirven para hacer predicciones de ‘una’ sabida ‘otra’; sin embargo, dos variables *dependientes*, SÍ pueden ser usadas para predecir una sabida otra.

Cuando se discuten relaciones entre tres variables aleatorias, aparece el concepto de independencia condicional, que se discute en el vídeo [961] y su continuación, el vídeo [962].

[951: indepdis]

Independencia estadística: discusión sobre interpretación, dificultades prácticas, etc.

**

13:33


Materiales: [independenciaestadisticaDiscusion.pdf]

[YouTube ▶]

Este vídeo discute en términos “filosóficos” el significado de la dependencia o independencia entre dos variables. Se argumentan las dificultades de determinar dicha dependencia basándose en muestras, y asimismo se hace hincapié en que dependencia estadística no significa que hayamos descubierto una relación “causa-efecto” (causalidad). Cuando se discuten relaciones entre tres variables aleatorias, aparece el concepto de independencia condicional, que se discute en el vídeo [961].

[952: tiger3]

Tigre oculto (3): Probabilidad condicional, marginal, fórmula de Bayes, interpretación gráfica

**

21:57

Materiales: [[CÓD.:](#) BayesNoRecursivoP1.mlx] [[PDF](#)]

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo continua la discusión sobre probabilidades marginales y condicionales que comenzamos en el vídeo [948], del que un rápido resumen es presentado en los primeros minutos del presente vídeo.

En particular, la relación entre marginales y condicionales será representada gráficamente en función de la probabilidad a priori... Si la probabilidad a priori de la condición es 1, marginal y condicional coinciden; si no es 1, entonces la marginal es una especie de “interpolación” de la condicional.


La segunda parte del vídeo discute la fórmula de Bayes para obtener la probabilidad condicional “inversa” (causa dado efecto) a partir de la conjunta o, mejor, como usualmente se escribe, a partir de la condicional “directa” (efecto dado causa) y la probabilidad a priori de la causa.

La probabilidad condicional de la causa dado cierto efecto se denomina también “a posteriori” si es evaluada después de efectivamente observar dicho efecto en una situación práctica.

Se representan diversas gráficas de probabilidad a priori y a posteriori para diferentes tablas condicionales que representan diferentes niveles de “precisión” en la observación del ruido de rugidos del tigre.

[953: idmidv]


Ampliación sobre dependencia estadística: Dependencia en media/homocedasticidad/correlación
Materiales: [indepcorelmediaindep.pdf]

***  14:21
 [YouTube ▶]

Este vídeo presenta dos casos en cierto modo “intermedios” entre, en un extremo, la independencia estadística y, en el otro extremo, el total determinismo. Estos casos son la “independencia en media” y la “independencia en varianza” (homocedasticidad). También se argumenta que la correlación 1 implica determinismo “lineal”, pero que pueden existir variables no correladas entre sí cuya relación sea (obviamente, no lineal) de hecho determinista.

[954: vamult]

Variables aleatorias multidimensionales (caso general): matriz de varianzas-covarianzas
Materiales: [vblesMultidimensionales.pdf]

***  08:44
 [YouTube ▶]

Este vídeo discute variables aleatorias multidimensionales: define los vectores de variables aleatorias, el vector de medias, y la matriz de varianzas-covarianzas. También discute cómo los cambios de variable lineales cambian las medias y la matriz de varianzas-covarianzas.

Presenta cómo calcular la matriz de varianzas-covarianzas muestral a partir de datos organizados en forma de matriz.

[955: vamdm]


Variables aleatorias multidimensionales: ejemplo Matlab
Materiales: [CÓD.: VariablesNDimensionalesExample.mlx] [PDF]

***  13:34
 [YouTube ▶]

Este vídeo presenta un ejemplo Matlab de análisis de 3 variables aleatorias, tanto un análisis marginal (histograma, media, varianza, sesgo, curtosis) como un análisis conjunto (matriz de varianzas-covarianzas, estandarización y matriz de correlación dos a dos, scatter-plot).

[956: dadonorm2d]

Distribución normal 2-dimensional: simulación de muestras (covarianza dada) en Matlab
Materiales: [CÓD.: TirarDadoNormal.mlx] [PDF]

**  09:16
 [YouTube ▶]

En este vídeo se muestra cómo usar el comando `randn` para obtener muestras de una variable normal multidimensional ψ con una cierta matriz de varianzas-covarianzas (vídeo [954]) dada de antemano Σ . La idea se basa en diagonalizar $\Sigma = VDV^T$, y hacer el cambio de variable $\phi = V^T\psi$ o, equivalentemente $\phi = V\psi$. Con ello, se comprueba que la matriz de varianzas-covarianzas de ϕ es diagonal (no correlada), que en distribución normal implica estadísticamente independiente y por tanto se puede “tirar el dado” con `randn` en cada componente de ψ y luego multiplicarlo por V para tener muestras de ψ , como se deseaba. Las variables ϕ se denominan “componentes principales” de ψ .

La parte final del vídeo comprueba cómo la gran mayoría de las muestras se encuentra dentro de ciertos elipsoides (alineados con los ejes, en el caso de los componentes principales ϕ , y “rotados” en el caso de las variables ψ). Los detalles sobre el cálculo de curvas de nivel para estos “elipsoides de confianza” se abordan en los vídeos [957] y [958], que usan la distribución de probabilidad χ^2 .

Nota: La diagonalización calcula la “raíz cuadrada” de la matriz de varianzas-covarianzas $V\sqrt{D}$, que es por lo que hay que multiplicar `randn`; esta raíz cuadrada de la matriz tiene la interpretación de “componentes principales” muy útil en muchas aplicaciones, pero existen

otras raíces cuadradas (factorización $\Sigma = QQ^T$), más rápidas computacionalmente como la factorización de Cholesky (comando `chol`), con Q triangular, que también son válidas para generar las muestras: si $\psi = Q \cdot s$ con $\text{cov}(s) = I$ entonces $E[\psi\psi^T] = Q \cdot I \cdot Q^T = \Sigma$. De hecho, en la documentación de Matlab (R2020b) de `randn` se utiliza la factorización de Cholesky para generar variables aleatorias normales multidimensionales en sus ejemplos.

Nota: Las variables aleatorias muestradas en diferentes instantes de *tiempo*, con posibles correlaciones entre las variables en diferentes instantes de tiempo se denominan *series temporales* o *procesos estocásticos*. Son la generalización estadística de los sistemas dinámicos deterministas. Dada su importancia en control, se les ha dedicado materiales específicos en el texto principal, ver Capítulo 7.

B.4 Intervalos (monovariable) y elipsoides (multivariable, normal) de confianza

[957: intc]
Intervalo/círculo de confianza en una o dos variables normales estándar

** 12:09

Materiales:

[ConfIntEsfera.pdf]

[YouTube ▶]

Este vídeo describe el intervalo de confianza $1 - \alpha$ en una distribución genérica, a partir del límite inferior (probabilidad acumulada de que sea *menor* de dicho límite igual a $\alpha/2$) y el límite superior (probabilidad acumulada de que sea *mayor* de dicho límite igual a $\alpha/2$). Con una función de densidad arbitraria, el cálculo de dichos límites requiere manipulaciones con la integral de la densidad (función de distribución). En el caso de la distribución normal estándar (media cero, desviación típica unidad), se puede usar `norminv` para calcularlos.

A continuación se plantea el caso de determinar una “región” de confianza para DOS variables normales estándar (y_1, y_2), considerando ambas $N(0, 1)$ independientes. Calculando intervalos de confianza de forma “separada” para cada una de las dos variables se puede calcular un “cuadrado” de confianza (el cuadrado 95% sería la intersección de las “bandas” generadas por los intervalos de confianza $\sqrt{0.95}$ en cada una de ellas).

Alternativamente, usando la distribución χ^2 (que se define como la distribución de probabilidad de sumas de cuadrados de normales estándar) se puede determinar el radio de un círculo dentro del cual el punto (y_1, y_2) se encontrará con una probabilidad prefijada p ; este radio será `sqrt(chi2inv(p,2))`. **Nota:** Este enfoque χ^2 es el que se utiliza en la mayor parte de las aplicaciones.

[958: elipc]
Elipsoide de confianza en distribución normal multidimensional: caso general

*** 08:22

Materiales:

[ConfEllipse.pdf]

[YouTube ▶]

Este vídeo generaliza las ideas del vídeo anterior [957] al caso de un vector de m variables aleatorias $y = (y_1, \dots, y_m)^T$ que se rigen por una distribución normal m -dimensional de media cero y varianza 1 independientes. Básicamente, se trata de sustituir los 2 grados de libertad del χ^2 del referido vídeo por m , esto es, usar `chi2inv(p,m)`. El caso 2D está discutido en detalle en los vídeos [959] y [960].

El vídeo analiza posteriormente el caso de distribución normal con correlación entre variables. Un ejemplo Matlab de dicha distribución y de cómo “tirar el dado” para obtener muestras de ella aparece en el vídeo [956], cuya visualización se aconseja de forma previa/simultánea a este contenido.

Si las variables siguen una distribución normal m -dimensional de media cero pero de matriz de varianzas-covarianzas Σ , esto es, tienen una función de densidad:

$$f(\mathbf{y}) = \frac{1}{(2\pi)^{m/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{y}^T \Sigma^{-1} \mathbf{y}\right)$$

entonces, a partir de una diagonalización de $\Sigma = VS^2V^T$ se obtienen unos “componentes principales normalizados” $\nu = \nu = S^{-1}V^T y$ que son variables de distribución normal $N(0, 1)$ independientes entre sí, por lo que se puede aplicar la fórmula χ^2 . Des haciendo el cambio de variable, resulta que la probabilidad de que la variable con distribución normal y esté dentro del elipsoide $y^T \Sigma^{-1} y \leq \text{chi2inv}(p, m)$ es p .

[959: cfrc1] **Distribución normal 2D: rectángulos y elipse de confianza**
(ejemplo Matlab)

Materiales: [[CÓD.:](#) IntConfVSchi2ESP.mlx] [[PDF](#)]

** 18:27

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo calcula el ‘círculo’ de máxima probabilidad por unidad de área para una distribución normal estándar 2D (esto es, con covarianza identidad), usando la fórmula χ^2 , ver vídeo [958]. También calcula regiones de confianza ‘marginales’ estudiando cada variable por separado (p.ej., $\pm 1.96\sigma$ daría la región 95%). Multiplicando las probabilidades de un intervalo se calcularían las probabilidades de un ‘rectángulo’ de confianza (cuadrado en este caso). La parte final del vídeo discute la elipse de confianza que saldría con una covarianza $\text{diag}([1 \ 4])$, y los rectángulos resultantes. Los cálculos si la matriz de varianzas-covarianzas no fuera diagonal no serían válidos; esto se discute con algo más de detalle en el vídeo [960], continuación de éste.

[960: cfrcwrong] **Distribución normal 2D: observaciones sobre rectángulos y elipse de confianza con correlación entre variables**

Materiales: [[CÓD.:](#) IntConfVSchi2ESP.mlx] [[PDF](#)]

*** 07:29

[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Este vídeo revisa las ideas principales del vídeo [959], y plantea la pregunta de si los cálculos de regiones de confianzas en dicho vídeo seguirían o no siendo correctos con variables normales correladas. Brevemente, se plantea que el elipsoide sí (aunque no estará alineado con los ejes, se discute la relación de la geometría de dicho elipsoide con los autovalores y autovectores de la matriz de varianzas-covarianzas). También los intervalos de confianza marginales serán correctos (las marginales de una normal 2D es una normal 1D). Sin embargo, los ‘rectángulos’ que se obtenían multiplicando las probabilidades de los intervalos de confianza NO darían un resultado de probabilidad correcto al no ser las variables marginales independientes. Para hacerlo correctamente, los rectángulos deberían estar alineados con los ejes de la elipse (componentes principales). Ese hecho de transformar las variables de una normal multidimensional (rotar) para que la matriz de covarianzas sea diagonal (elipsoides de confianza alineados con los ejes) es lo que motiva lo que se denomina análisis de componentes principales en estadística multivariante, ver vídeos [710] y [711]. También puedes visualizar el vídeo [958] con más detalle sobre el elipsoide de confianza normal multivariante.

Independencia condicional, redes bayesianas

[961: condin1] **Independencia condicional (I): definición y ejemplos básicos**

Materiales: [conditionalindepES.pdf]

*** 12:53


[YouTube ▶]

* [LINK TO ENGLISH VERSION](#)

Las ideas de independencia estadística (videos [950], [951]) pueden refinarse para hablar de la independencia de las distribuciones de probabilidad condicional (vídeo [946]). Con ello pueden definirse las variables aleatorias “condicionalmente independientes dado cierto valor de otra tercera variable”. Aunque pueda parecer un refinamiento inútil de interés para muy poca gente, éste no es el caso: hay detrás ideas muy relevantes de “sentido común”, y está detrás de las ideas de “estado” en Física y Teoría de Control y las “redes bayesianas” en Inteligencia Artificial.

Este vídeo discute la definición teórica y unos ejemplos “intuitivos” para comprender el concepto (marca de coches versus multas; altura versus titulación académica, etc.). El segundo vídeo [962], continuación de éste, discute su relación con los modelos de sistemas dinámicos usados en control y las redes bayesianas usadas también en otros ámbitos de informática e Inteligencia Artificial.

Para comprender la idea en un problema muy simple, puedes visualizar el vídeo [600] parte del caso de estudio ‘tigre oculto’ que utilizamos para ilustrar conceptos básicos de probabilidad. Ejemplos adicionales (particularizados al caso de correlación o no correlación *condicional*) se discuten en el vídeo [963], cuya visualización se aconseja.

[962: condin2]	Independencia condicional (II): ejemplos adicionales relevantes en control (Markov), redes Bayesianas Materiales: [conditionalindepES.pdf]	***  16:30 [YouTube ►]
----------------	--	--


*LINK TO ENGLISH VERSION

El vídeo [961] definió el concepto de variables aleatorias “condicionalmente independientes dado cierto valor de otra tercera variable”. Aunque pueda parecer un refinamiento inútil de interés para muy poca gente, éste no es el caso: hay detrás ideas muy relevantes de “sentido común”, y está detrás de las ideas de “estado”, fundamental en teoría de control multivariable, y las “redes bayesianas” en Inteligencia Artificial.

Este vídeo complementa los ejemplos “intuitivos” del vídeo arriba referido con ejemplos más abstractos sobre ruidos de medida y de proceso, el concepto de estado, hipótesis de Markov, y los diagramas de redes bayesianas asociados. Se recomienda visualizar el vídeo [10] para más discusiones sobre la importancia del concepto de estado y Markov en el modelado de sistemas físicos.

También presenta conclusiones sobre la importancia del concepto de independencia condicional en el pensamiento científico en general.

Ejemplos adicionales (particularizados al caso de correlación o no correlación *condicional*) se discuten en el vídeo [963], cuya visualización se aconseja, y en el caso de un tigre oculto en vídeo [600].

[963: condnoco]	Independencia condicional (III): variables aleatorias condicionalmente NO correladas (solo distr. normal multivariable) Materiales: [Cód.: conditionalnocorrelaexample1.mlx] [PDF]	***  16:59 [YouTube ►]
-----------------	--	--

*LINK TO ENGLISH VERSION

Este vídeo complementa la discusión sobre independencia condicional del vídeo [961], particularizándola al caso donde sólo se desean hacer predicciones *lineales* y, por tanto, se debe analizar únicamente la matriz de varianzas-covarianzas.

Por tanto, la ausencia de covarianza (esto es, la no correlación) entre los errores de predicción de dos variables aleatorias dada una tercera es lo que definirá la no-correlación condicional... o algo así... o sea, nos tomamos “informalmente” los conceptos aquí lanzados, dado que sólo serían válidos ante el caso de que las variables estuviéramos seguros de que han sido generadas con una distribución normal multidimensional. Pero,

de hecho, como no-correlación en variables normales multidimensionales implica independencia pues no estamos viendo en este vídeo nada diferente del propio concepto de independencia condicional, si nos ponemos formalmente rigurosos.

La cuestión es que esa no-correlación condicional permite hacer redes bayesianas que “encadenan” modelos lineales con ruido aditivo, como se ve en los ejemplos numéricos.

B.5 Predicción estadística

[964: pre1] Predicción estadística: generalidades

Materiales: [PrediccionEstadistica.pdf]

**  11:00
[YouTube ▶]

Este vídeo discute las generalidades del concepto de “predicción” en estadística. Idealmente una predicción sería cualquier metodología de cálculo o de experimentación que permitiera aproximar la función de densidad $f(y)$ de una variable aleatoria y .

No obstante, la función de densidad completa resulta complicada de estimar y de manejar posteriormente. En muchas aplicaciones se utiliza una predicción “puntual”: un número “ p ” que resume características de interés sobre la distribución de probabilidad modelada con $f(y)$. Las predicciones puntuales más extendidas son la *moda* (valor más probable, *maximum likelihood prediction* en la literatura en inglés) o la *media* (en variables que toman valores reales, la media –si la función de densidad es simétrica alrededor de ella– es el valor tal que existe un 50% de probabilidad de que y que sea menor, 50% de probabilidad de que y sea mayor o igual; esta interpretación no es cierta en distribuciones asimétricas con skewness $\neq 0$: en ese caso, a ese valor 50% se le denomina “mediana”).

La media también puede ser entendida como el valor constante ξ que minimiza $E[(y - \xi)^2]$, una especie de mínima “varianza” (no exactamente, porque varianza implica restar la media). Podría también pensarse en predicciones puntuales que solucionaran un problema de optimización distinto a la mínima varianza.

Un intermedio entre las predicciones completas (función de densidad, histograma “fino”), sería dar la media y un intervalo de confianza donde se situen la mayor parte (usualmente 95% o 99%) de las posibles observaciones de y .

Al final del vídeo, se comenta en el interés de poder actualizar una predicción si se dispone de información sobre otras variables “relacionadas” con la variable a predecir. Definir y discutir esa idea será objeto del vídeo [965] y sucesivos.

[965: preco] Predicción condicional de una variable aleatoria dada otra estadísticamente dependiente

Materiales: [PredCondicional.pdf]

***  11:00
[YouTube ▶]

Este vídeo define los conceptos asociados a la predicción de una variable y dada otra x (información, que se supone es estadísticamente dependiente, en algún modo, de y): básicamente la mejor predicción sería la función de densidad condicional $f(y|x)$. Con un modelo de la función de densidad conjunta $f(x, y)$, podríamos decir $f(y|x) = \frac{f(x, y)}{\int f(x, y) dx}$, pero en general la densidad condicional resulta de difícil cálculo y manejo. Por ello, como en el vídeo [964], se plantea la conveniencia de utilizar predicciones condicionales *puntuales*. La moda o la media de dicha densidad condicional son los obvios candidatos.

En concreto, se propone como predicción óptima puntual a la media condicional, esto es $E[y|x] = \int y f(y|x) dy$. Tiene, como propiedad de interés, el ser aquella que minimiza $E[(y - \xi(x))^2]$ para toda posible función $\xi(x)$. No obstante, se argumenta que este concepto es, también, difícil de calcular y manejar (experimentalmente se requerirían “muchas” muestras a promediar para cada valor de x , lo cual es imposible en variables x reales).

Por ello, en la práctica se ajustan funciones más sencillas, como por ejemplo la “regresión lineal”, cuya interpretación en términos estadísticos es el objetivo de los vídeos [966] a [968].

[966: preli1] Predicción lineal óptima: introducción y definición

Materiales: [PrediccionLineal.pdf]

***  09:07

[YouTube ►]

Este vídeo define los conceptos asociados a la “predicción lineal óptima” de una variable y en función de otra x (información), en términos estadísticos. Dicha predicción se expresa con una fórmula $y = \Theta x + \mu$, siendo Θ y μ parámetros constantes (o matrices constantes, en casos multivariable). Tiene las propiedades siguientes:

1. El error $e := y - p(x)$ tiene media cero.
2. El error y la información x no tienen *correlación* (covarianza $E[ex^T] = 0$).
3. Minimiza $E[(y - \Theta x - \mu)^2]$, esto es, relacionado con la mínima varianza del error de predicción de entre todos los predictores lineales.

La predicción lineal óptima coincide con el ajuste determinista de mínimos cuadrados. La diferencia entre la predicción lineal óptima y la media condicional $E[y|x]$ es el *error sistemático* del modelo lineal, que en la mayoría de los casos es demasiado simple para poder ajustar la media condicional de datos generados mediante modelos no-lineales sujetos a ruido.

[967: preli2] Predicción lineal óptima: fórmulas basadas en matriz de varianzas-covarianzas

Materiales: [FormulasPredOptimaLineal.pdf]

***  13:55

[YouTube ►]

Este vídeo demuestra que, dadas dos variables aleatorias x, y con medias μ_x, μ_y respectivamente, y con matriz de varianzas covarianzas:

$$E[(y \ x)] := \begin{pmatrix} \Sigma_y & \Sigma_{yx} \\ \Sigma_{yx}^T & \Sigma_x \end{pmatrix}$$

la mejor predicción lineal viene dada por:

$$p(x) = \Sigma_{yx} \Sigma_x^{-1} (x - \mu_x) + \mu_y$$

Además, la varianza del error de predicción *a posteriori* viene dada por:


$$\Sigma_e = \Sigma_y - \Sigma_{yx} \Sigma_x^{-1} \Sigma_{yx}^T$$

Con ello, se justifica que si $\Sigma_{yx} = 0$ (variables no correladas) entonces la mejor predicción lineal es la media de y (exactamente igual al caso de no tener información), y no se reduce la incertidumbre, pero que si existe correlación entre x e y , la información permite refinar el estimado “a posteriori” y reducir la incertidumbre sobre y respecto al caso de no tener información: formalmente, $\Sigma_e < \Sigma_y$ si Σ_x está acotada y $\Sigma_{yx} \neq 0$.

Nota: que la mejor predicción de y dado x sea, por ejemplo $p_{y|x} = 2x$ no implica que la mejor predicción de x dado y sea $p_{x|y} = 0.5y$. El modelo “inverso” en sentido estadístico no coincide en la mayor parte de casos con el modelo inverso algebraico (el ruido hace que se “pierda información”, que aumente la entropía, etc.). Estas ideas son desarrolladas en los vídeos [969] y [970].

[968: prelim] Predicción lineal óptima: ejemplo Matlab

Materiales: [Cód.: MejorPredLineal.mlx] [PDF]

**  10:11


[YouTube ►]

En este vídeo se presenta un ejemplo Matlab donde se estima la matriz de varianzas-covarianzas de un par de variables a partir de una nube de datos, con el comando `cov`, y se construye el predictor lineal óptimo a partir de dicha matriz (que, obviamente, se comprueba coincide con los mínimos cuadrados deterministas obtenidos mediante pseudoinversa `pinv`). Se representa la recta de regresión y los intervalos de confianza $\pm 2\sigma$ (*a posteriori*), comprobándose que, en efecto, la desviación típica estimada *a posteriori* es un 60% de la desviación de y sin tener en cuenta la información de x . Si los datos tuvieran distribución normal, esos intervalos de confianza por encima y debajo de la recta de regresión deberían contener el 95% de los datos... aunque en una situación práctica no se puede, en general, garantizar que la hipótesis de “distribución normal” se cumple en el mecanismo que genere los datos (puesto que este mecanismo es desconocido, en general).

El concepto de predicción estadística sobre variables con correlación en distintos instantes de tiempo (series temporales) da lugar a lo que se conoce como Filtro de Kalman. Es un desarrollo de importancia central en teoría de control, discutido en la Sección 17 del texto principal.

[969: vcinv1]

Mejor predicción lineal: modelo con ruido aditivo, modelo inverso en sentido estadístico, teoría y ejemplo

***  18:25

Materiales: [[CÓD.:](#) mejorpredlinealversusmodelo.mlx] [[PDF](#)]

[YouTube ▶]

*[LINK TO ENGLISH VERSION](#)

Este vídeo ilustra cómo a partir de una matriz de varianzas-covarianzas dada de un par de variables aleatorias (a, b) , aplicando la fórmula de mejor predicción lineal (vídeo [967]), puede expresarse dicha predicción como un modelo lineal con ruido aditivo independiente por ejemplo $b = \theta a + \epsilon$, siendo $\theta = \Sigma_{ba} \Sigma_{aa}^{-1}$ y ϵ de varianza $\Sigma_{bb} - \Sigma_{ba} \Sigma_{aa}^{-1} \Sigma_{ba}^T$. Ello no quiere decir que la “realidad física” siga esa expresión lineal, pero esa expresión explica la varianza observada.


También se aborda el proceso inverso: dado un modelo lineal con ruido $b = \theta a + \epsilon$, se obtiene la matriz de varianzas covarianzas asociada.

La última parte del vídeo discute el hecho de que dada una matriz de varianzas-covarianzas, el estimado de “b dado a”, $b = \theta a + \epsilon$ es diferente de invertir algebraicamente la ecuación del estimado de “a dado b” ($a = \eta b + \epsilon'$), esto es, $\eta \neq \theta^{-1}$. Se representa gráficamente la diferencia entre ambos modelos.

Otro ejemplo sobre las ideas aquí discutidas (inverso estadístico de mínima varianza versus inverso algebraico) aparece en el vídeo [970], continuación de éste.

[970: vcinv2]

Mejor predicción lineal: modelo inverso en sentido estadístico, ejemplo adicional

***  10:17

Materiales: [[CÓD.:](#) mejorpredlinealversusmodeloInvers.mlx] [[PDF](#)]

[YouTube ▶]


*[LINK TO ENGLISH VERSION](#)

Este vídeo ilustra las ideas del vídeo [969] sobre la estimación de un modelo “inverso” de otro con un ejemplo rápido adicional. Se comprueba que el inverso “estadístico” no es igual al algebraico salvo que la relación señal-ruido sea muy buena (varianza ruido tienda a cero).

B.5.1 Interpretaciones deterministas y estadísticas de las soluciones del problema de mínimos cuadrados

[971: ls3i]

Tres interpretaciones de los mínimos cuadrados: optimización, proyección, estadística

****  13:48

Materiales: [TresInterprMinimosCuadrados.pdf]

[YouTube ▶]

Este vídeo discute las tres interpretaciones posibles para la misma fórmula de mínimos cuadrados (pseudoinversa):

1. Interpretación como solución a un problema de **optimización determinista** (ajuste de parámetros),
2. Interpretación **geométrica** (pseudoinversa, proyección ortogonal –vídeo [914]–),
3. Interpretación **estadística** (mínima varianza, mínima correlación de error).

Dependiendo del contexto se usará una interpretación u otra de la misma fórmula.

[972: ls3im]

Tres interpretaciones de los mínimos cuadrados (optimización, proyección, estadística): caso multivariable (matricial)

☰
14:42

Materiales:

[TresInterprMinimosCuadradosMIMO.pdf]

[YouTube ▶]

Este vídeo plantea las tres interpretaciones anteriores (vídeo [971]) sobre un problema multivariable (mínimos cuadrados matriciales) de obtener un modelo de ajuste/predicción lineal de un vector $y \in \mathbb{R}^p$ a partir de información $x \in \mathbb{R}^m$, dadas N muestras de y y de x . El resultado es una *matriz* $\Theta \in \mathbb{R}^{p \times m}$ que se obtiene con fórmulas de pseudoinversa y que conserva la misma interpretación que el caso discutido en el vídeo [971].

El contenido de este vídeo es complementario al vídeo [755] que se recomienda visualizar conjuntamente con este.