Control mediante discretización de reguladores contínuos: Conclusiones finales

Antonio Sala

Control de Sistemas Complejos

DISA – Universitat Politècnica de València

Video-presentación disponible en:

http://personales.upv.es/asala/YT/V/**drefin**.html

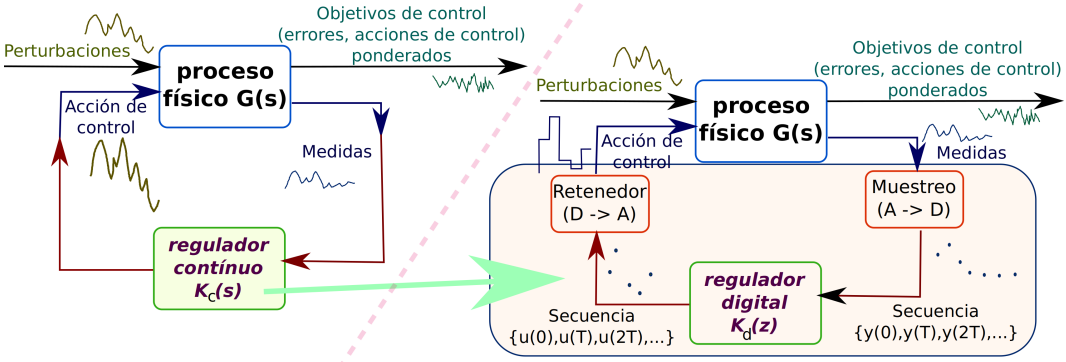
Introducción

Motivación:
Estabilidad de la discretización (integración numérica) no garantiza conservar estabilidad o prestaciones en bucle cerrado debido a la aproximación, salvo que $T \to 0$. Muchas reglas y opciones para la discretización... ¡quizás demasiadas!.

Objetivos:
Comprender y sintetizar las ideas clave de las distintas opciones.

Contenido:
Discusión/crítica sobre selección del período de muestreo, diseño del regulador contínuo y metodología.

Discretización de reguladores contínuos



1. Diseño inicial en tiempo contínuo.
2. **Aproximación** de $K(s)$ por $\mathcal{K}:=\rightarrow[\text{Muestreo}]\rightarrow K(z)\rightarrow[\text{reten.}]\rightarrow$

¿Discretización exacta o aproximada?

La discretización de un regulador $K_c(s)$ no es exacta como sí lo era la ZOH para procesos (para diseño discreto $K_d(z)$ posterior).

No existen garantías "al 100%" de que si el bucle cerrado contínuo es estable y tiene unas ciertas prestaciones, que también las tenga el bucle por computador.

Único diseño exacto a período grande (estabilidad garantizada para cualquier T): diseñar $K_c(s)$ para el proceso $\tilde{G}$=`d2c(c2d(G,T,'zoh'),T,'tustin')`, y discretizar K_c por Tustin –es equivalente a un diseño directo discreto–.

*La resp. frecuencia $[0,\infty]$ se comprime a $[0,\pi/T]$, deforma respuesta temporal también.

Discretización *versus* simulación (¿?)

- Simulación (integración numérica) y discretización son **básicamente lo mismo**, con matices en la terminología...

- Simulación/integración numérica suele ser utilizado en contextos **no lineales** para obtener de forma aproximada la respuesta temporal.
- "Discretización" de reguladores enfatiza el problema de control por computador (presencia de muestreo/retenedor, sistema híbrido contínuo-discreto). Es *aproximada* incluso con reguladores lineales.
- "Discretización" de procesos es exacta (ZOH) o "casi" (FOH, exacta con entradas lineales entre muestras), en ese caso se usan fórmulas exponenciales.

Selección del período de muestreo

- Reglas basadas en Tma. Shannon (ancho de banda de referencias, perturbaciones)
- Reglas "de andar por casa" (basadas en tiempo de establecimiento, de subida, frec. propia, ...)
- Reglas basadas en estabilidad numérica de la discretización del regulador
- Reglas basadas en la distorsión de la respuesta en frecuencia de bucle cerrado

La selección del período es un poco "por prueba y error", ante la duda mejor muestrear más rápido... pero sin pasarse (problemas de cuantización/precisión numérica/identificación/tiempo de cómputo).

¿El diseño continuo está bien hecho?

Cuando todas las reglas de selección de T dicen algo parecido, eso es "buena señal"... Si no, quizás regulador continuo mal diseñado:

- ¿Para un proceso con $t_{est} = 1$ s, polo dominante en $s = -4$, filtro de ruido $1/(0.0001s+1)$?
  - ¿Seguro que no lo ponemos a $1/(0.02*s+1)$, polo filtro en $s = -50$?
  - ¿Y si ese filtro de ruido lo hacemos formar parte del anti-aliasing? (se implementa en continuo, RC+operacional; incorporar en offset+amplificación analógica antes de A→D)
- ¿estamos pidiendo "demasiado" en el diseño de $K_c(s)$?
  ¿Estamos cambiando "demasiado" la respuesta en bucle abierto, y requerimos controladores complejos, de alta ganancia a alta frecuencia y polos muy rápidos? ¿será sensible a errores de modelado (compromiso especificaciones-robustez)?... necesitaremos período muy pequeño para discretizarlos, y en la práctica darán problemas (controladores poco "robustos").

Metodología

1. Antes de discretizar:
   - Considerar añadir $(1 - Ts/2)$ para contemplar efecto inestabilizante del muestreo, o/y un polo rápido de los filtros anti-alias/reconstrucción si va a haber.
   - Identificar el rango de "frecuencias" importante para el funcionamiento del bucle, a partir de Bode de $S = (1 + GK)^{-1}$, $T = GK/(1 + GK)^{-1}$, $KS = K(1 + GK)^{-1}$.
   - Reducir el orden del regulador (eliminar polos a frecuencias "demasiado altas", integrarlos en el anti-alias analógico).
2. Elegir el período basándose en las reglas mencionadas, ante la duda, escoger el muestreo más rápido (sensatamente).
3. Discretizar... ¿método? ante la duda, elegir Tustin/foh, sobre todo si se ha añadido $(1 - Ts/2)$ para diseñar $K_c(s)$, aunque Euler (FW/BW) funcionará bien si no hay polos de regulador con $|p| > 1/T$ (FW), o poco amortiguados, o inestables (BW).

Nota: Tustin no funciona "siempre" mejor que Euler.

Todo es muy orientativo...

- Muestrear rápido es barato... en monovariable, una vez hecha la compra de hardware de 20 KHz, 16 bit ADC, red 1 GBit/s, latencia 15 μs.
- ¿Filtro anti-alias (RC+operacional analógico) y $T = 0.2$ s? ¿O $T = 0.01$ s y me olvido del anti-alias –filtro digital–?
- Muestrear una fábrica entera a $T = 0.0005$ s es inviable. Control jerárquico en cascada (bucles internos servo actuadores/smart sensor $T \leq 50$ ms, PIDs industriales $0.05 < T < 2$ s, bucle externo control predictivo $T > 2$ s) suele ser habitual.
- La selección de T se puede hacer por prueba y error.
- Un PID analógico inicial puede discretizarse a período "relativamente grande" y hacer refinado "manual" a posteriori de K_p, K_d, K_i y cte tiempo filtro ruido.
- A veces, te venden un PID industrial con $T = 150$ ms fijo, y punto.

Conclusiones

- Muchas reglas de selección de T que pueden recomendar valores bastante diferentes.
- Muchos métodos de discretización (EulerFW, EulerBW, Tustin, foh).
- Prácticamente nada es exacto/garantizado (excepto $\hat{G}$=`d2c(c2d(G,T,'zoh'),T,'tustin')` y sólo "estabilidad").
- Al final, los "errores de discretización" son uno más de las muchas fuentes de error de modelado en aplicaciones. Si se sospecha que son importantes, diseño directo discreto (prueba y error, asign polos, óptimo/predictivo, ...), idealmente multifrecuencia o "sampled-data".

En resumen, en una aplicación práctica de bajo orden con $t_{est} = 3$ s y PIDs sencillos, todo se puede hacer funcionar "parecido al continuo" con $0.001 \leq T \leq 0.1$ y un poco de "maña"... con $T > 0.3$ empieza a no ser tan fácil parecerse al diseño continuo.