Modelado para control robusto: alternativas con distinto numero de incidencias de un párametro incierto

Presentación en vídeo: http://personales.upv.es/asala/YT/V/urealmult.html

Este código funcionó correctamente con Matlab **R2020a**

Objetivo: diferentes formas de escribir las ecuaciones de un modelo dinámico con parámetros inciertos pueden llevar a realizaciones no mínimas (que solventaríamos con `minreal`) pero también a diferente número de veces (incidencias) que aparece un determinado parámetro incierto. El comando `simplify` de la Robust Control Toolbox intenta reducir dicho número de incidencias, pero todo resulta más fácil si las ecuaciones de la física son agrupadas de modo que el parámetro incierto aparece el menor número de veces posible.

*Nota: en versiones posteriores de Matlab, es posible que sea más "inteligente" con el tema multiincidencia y salgan menos "occurrences" que en este mlx/PDF guardado.

Tabla de Contenidos

Modelo (simbólico) de un péndulo linealizado

```
syms dx1dt dx2dt x1 x2 M L F g b
modelo=[dx1dt==x2;M*L^2*dx2dt==-sin(x1)*L*M*g-b*x2+F]
```

```
modelo =
```

$$\begin{pmatrix} \mathrm{dx1dt} = x_2 \\ L^2\, M\, \mathrm{dx2dt} = F - b\, x_2 - L\, M\, g\, \sin(x_1) \end{pmatrix}$$

```
EcEstado=solve(modelo,{dx1dt,dx2dt});
dxdt=[EcEstado.dx1dt; EcEstado.dx2dt]
```

```
dxdt =
```

$$\begin{pmatrix} x_2 \\ -\dfrac{b\, x_2 - F + L\, M\, g\, \sin(x_1)}{L^2\, M} \end{pmatrix}$$

Para linealizar la ecuación de estado hacemos derivadas parciales:

```
Asym=jacobian(dxdt,[x1 x2])
```

```
Asym =
```

$$\begin{pmatrix} 0 & 1 \\ -\frac{g\cos(x_1)}{L} & -\frac{b}{L^2 M} \end{pmatrix}$$

```
Bsym=jacobian(dxdt,F)
```

```
Bsym =
```

$$\begin{pmatrix} 0 \\ \frac{1}{L^2 M} \end{pmatrix}$$

Sustituimos ahora por valores en punto de equilibrio:

```
x1eq=0;
Asym_eq=subs(Asym,x1,x1eq)
```

```
Asym_eq =
```

$$\begin{pmatrix} 0 & 1 \\ -\frac{g}{L} & -\frac{b}{L^2 M} \end{pmatrix}$$

En B no hace falta, porque no sale x1.

Sustitución de valores de parámetros (conocidos o inciertos)

Hacemos los parámetros constantes o inciertos:

```
g=9.81;
b=ureal("b",0.7,"PlusMinus",[-.1,.1]);b.AutoSimplify='full';
L=ureal("L",1.25,"Percent",2);L.AutoSimplify='full';
M=ureal("M",0.5,"Percent",5);M.AutoSimplify='full';
```

Nota importante: estamos incorporando la incertidumbre paramétrica, pero no la debida al error de linealización.

Los sustituimos en las expresiones simbólicas para obtener objetos de la robust control toolbox:

```
A=evalin('base',char(Asym_eq))
```

```
A =

  Uncertain matrix with 2 rows and 2 columns.
  The uncertainty consists of the following blocks:
    L: Uncertain real, nominal = 1.25, variability = [-2,2]%, 3 occurrences
    M: Uncertain real, nominal = 0.5, variability = [-5,5]%, 1 occurrences
    b: Uncertain real, nominal = 0.7, variability = [-0.1,0.1], 1 occurrences

Type "A.NominalValue" to see the nominal value, "get(A)" to see all properties, and "A.Uncertainty" to int
```

```
A.NominalValue
```

```
ans = 2×2
         0    1.0000
```

```
   -7.8480   -0.8960
```

```
B=evalin('base',char(Bsym))
```

```
B =

  Uncertain matrix with 2 rows and 1 columns.
  The uncertainty consists of the following blocks:
    L: Uncertain real, nominal = 1.25, variability = [-2,2]%, 2 occurrences
    M: Uncertain real, nominal = 0.5, variability = [-5,5]%, 1 occurrences

Type "B.NominalValue" to see the nominal value, "get(B)" to see all properties, and "B.Uncertainty" to int
```

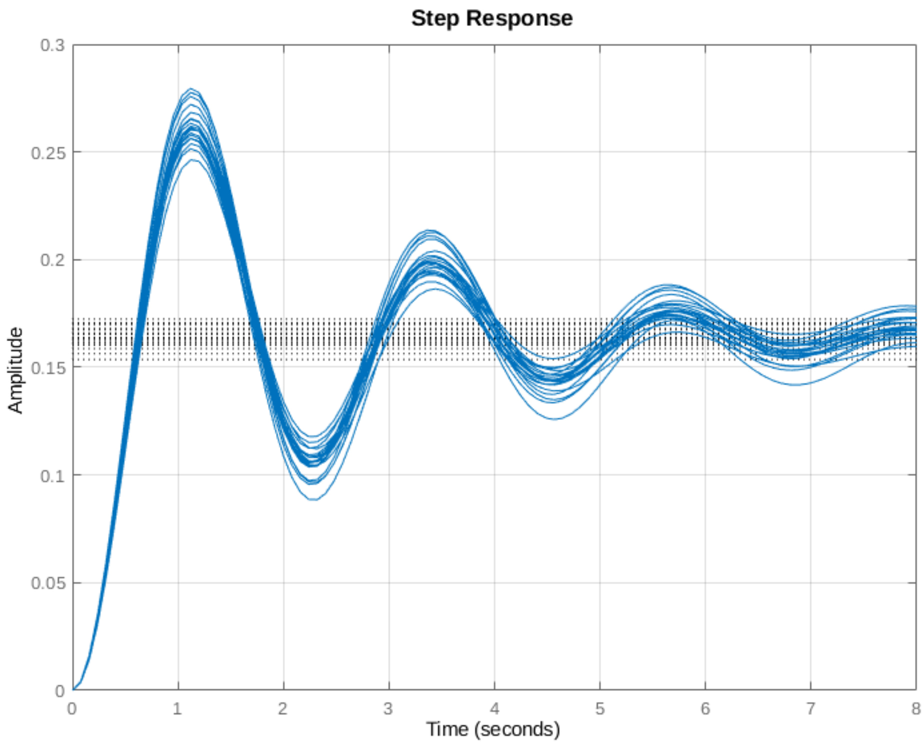
```
B.NominalValue
```

```
ans = 2×1
         0
    1.2800
```

```
sys=ss(A,B,[1 0],0) %escogemos posición angular como salida
```

```
sys =

  Uncertain continuous-time state-space model with 1 outputs, 1 inputs, 2 states.
  The model uncertainty consists of the following blocks:
    L: Uncertain real, nominal = 1.25, variability = [-2,2]%, 5 occurrences
    M: Uncertain real, nominal = 0.5, variability = [-5,5]%, 2 occurrences
    b: Uncertain real, nominal = 0.7, variability = [-0.1,0.1], 1 occurrences

Type "sys.NominalValue" to see the nominal value, "get(sys)" to see all properties, and "sys.Uncertainty"
```

```
step(sys,8), grid on
```



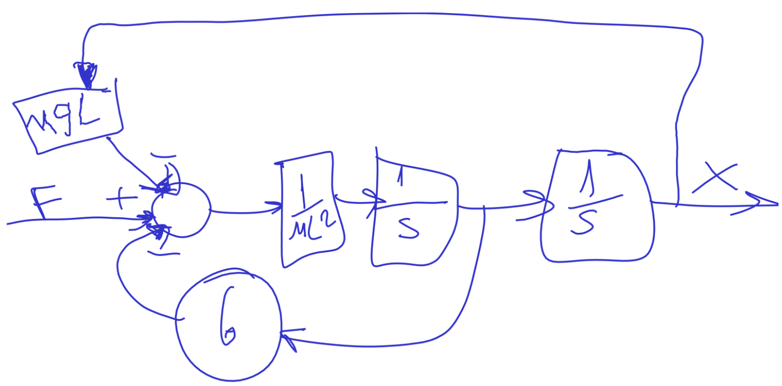
Intentamos reducir (sin éxito) el número de incidencias de alguno de los parámetros:

```
simplify(sys,'full')
```

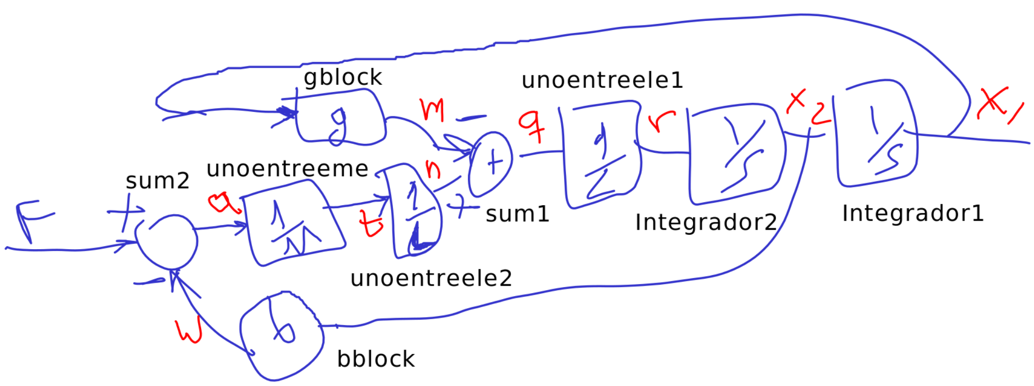
```
ans =

  Uncertain continuous-time state-space model with 1 outputs, 1 inputs, 2 states.
  The model uncertainty consists of the following blocks:
    L: Uncertain real, nominal = 1.25, variability = [-2,2]%, 5 occurrences
    M: Uncertain real, nominal = 0.5, variability = [-5,5]%, 2 occurrences
    b: Uncertain real, nominal = 0.7, variability = [-0.1,0.1], 1 occurrences

Type "ans.NominalValue" to see the nominal value, "get(ans)" to see all properties, and "ans.Uncertainty"
```

Alternativa: forma "manual" a partir de diagrama de bloques linealizado

Podemos considerar este diagrama de bloques, con 2 incidencias de "M", y 3 de "L", y una de "b":



Pero podemos transformarlo en este otro con 1 incidencia de "M", dos de "L" y 1 de "b":




```
s=tf('s');
integrador1=1/s; integrador1.InputName={'x2'};integrador1.OutputName={'x1'};
integrador2=1/s; integrador2.InputName={'r'};integrador2.OutputName={'x2'};
unoentreele1=tf(1/L);unoentreele1.InputName={'q'};unoentreele1.OutputName={'r'};
sum1=sumblk('q=n-m');
unoentreele2=tf(1/L);unoentreele2.InputName={'t'};unoentreele2.OutputName={'n'};
gblock=tf(g);gblock.InputName={'x1'};gblock.OutputName={'m'};
```

```
unoentreeme=tf(1/M);unoentreeme.InputName={'a'};unoentreeme.OutputName={'t'};
sum2=sumblk('a=F-w');
bblock=tf(b);bblock.InputName={'x2'};bblock.OutputName={'w'};
sys2=connect(integrador1,integrador2,unoentreele1,unoentreele2,sum1,gblock,unoentreeme,
```

```
sys2 =

  Uncertain continuous-time state-space model with 1 outputs, 1 inputs, 2 states.
  The model uncertainty consists of the following blocks:
    L: Uncertain real, nominal = 1.25, variability = [-2,2]%, 2 occurrences
    M: Uncertain real, nominal = 0.5, variability = [-5,5]%, 1 occurrences
    b: Uncertain real, nominal = 0.7, variability = [-0.1,0.1], 1 occurrences

Type "sys2.NominalValue" to see the nominal value, "get(sys2)" to see all properties, and "sys2.Uncertaint
```

```
norm(sys2.NominalValue-sys.NominalValue,'inf')
```

```
ans = 2.2635e-17
```

Comparación de prestaciones/tiempo de cómputo con la Robust Control Toolbox

```
opt = robOptions('Display','on','Sensitivity','on');
tic,[rs,wcus]=robstab(sys,opt),toc
```

```
Computing peak...  Percent completed: 100/100
System is robustly stable for the modeled uncertainty.
 -- It can tolerate up to 650% of the modeled uncertainty.
 -- There is a destabilizing perturbation amounting to 700% of the modeled uncertainty.
 -- This perturbation causes an instability at the frequency 2.82 rad/seconds.
 -- Sensitivity with respect to each uncertain element is:
      4% for L. Increasing L by 25% decreases the margin by 1%.
      13% for M. Increasing M by 25% decreases the margin by 3.25%.
      88% for b. Increasing b by 25% decreases the margin by 22%.
rs = struct with fields:
           LowerBound: 6.4960
           UpperBound: 7.0000
    CriticalFrequency: 2.8222
wcus = struct with fields:
    L: 1.2317
    M: 0.6727
    b: -8.3045e-14
Elapsed time is 5.598353 seconds.
```

```
tic,[rs,wcus]=robstab(sys2,opt),toc
```

```
Computing peak...  Percent completed: 100/100
System is robustly stable for the modeled uncertainty.
 -- It can tolerate up to 658% of the modeled uncertainty.
 -- There is a destabilizing perturbation amounting to 700% of the modeled uncertainty.
 -- This perturbation causes an instability at the frequency 2.73 rad/seconds.
 -- Sensitivity with respect to each uncertain element is:
      0% for L. Increasing L by 25% decreases the margin by 0%.
      11% for M. Increasing M by 25% decreases the margin by 2.75%.
      90% for b. Increasing b by 25% decreases the margin by 22.5%.
rs = struct with fields:
           LowerBound: 6.5809
           UpperBound: 7.0000
    CriticalFrequency: 2.7269
wcus = struct with fields:
```

```
    L: 1.3192
    M: 0.4352
    b: 1.4433e-15
Elapsed time is 1.824625 seconds.
```

Conclusiones

Repeticiones innecesarias de los parámetros inciertos causan mayor tiempo de cómputo y resultados más conservativos cuando la Robust Control Toolbox se usa para analizar estabilidad robusta o diseñar controladores (**`musyn`**). Es, pues, muy recomendable evitar dichas repeticiones innecesarias en la fase de modelado.

Recalcamos de nuevo que la incertidumbre no lineal no ha sido modelada: en la documentación de Matlab existe el bloque **`udyn`** de incertidumbre no-lineal, variante en el tiempo, pero las rutinas de incertidumbre estructurada como robstab no lo manejan.