

Descripción externa e interna

Descripción interna de sistemas

Table of Contents

Descripción interna de sistemas.....	1
1. Modelado en el Espacio de Estados.....	1
1.1 El comando ss.....	2
1.2 Ejemplo descripción interna: Circuito LRC serie.....	3
1.3 Puntos de equilibrio.....	10

La **teoría de control clásica** trabaja con sistemas SISO y LTI, emplea métodos de respuesta en frecuencia y el lugar de las raíces, y conduce a sistemas estables que satisfacen un conjunto de requisitos pero que no son óptimos desde ningún punto de vista.

No obstante, las plantas modernas presentan cada vez una mayor complejidad, debido a que realizan cada vez tareas más complejas y de mayor precisión. La necesidad de controlar este tipo de plantas, que típicamente presentan múltiples entradas y salidas (MIMO), y pueden ser no lineales y variantes en el tiempo, junto con la disponibilidad de computadoras digitales, impulsó allá por 1960 el desarrollo de la **teoría de control moderna**. Esta teoría considera la **descripción interna** de los sistemas a controlar en lugar de la descripción externa empleada por la teoría clásica. Veamos en qué consiste.

1. Modelado en el Espacio de Estados

La descripción interna de sistemas se basa en el concepto de **estado**: conjunto más pequeño de **variables** (llamadas **de estado**) de forma que si se conoce su valor en $t = t_0$ y la(s) entrada(s) al sistema para $t \geq t_0$, estos determinan completamente la evolución de dichas variables de estado y la(s) función(es) de salida mediante las ecuaciones:

- Ecuación de estado: $\dot{x}(t) = f(x(t), u(t))$
- Ecuación de salida: $y(t) = g(x(t), u(t))$

donde:

- x es **vector de estado**, el cual contiene las recién presentadas variables de estado, y refleja el estado del sistema en un instante de tiempo,
- $f(\cdot)$ es la **función de transición**, la cuál permite conocer la evolución en función del estado actual y la entrada, y
- $g(\cdot)$ es la **función de salida**, que permite conocer la salida del sistema en función del estado y de su entrada.

Aunque se puede definir un espacio de estados que no tenga una relación evidente con el funcionamiento del sistema, es conveniente que las variables de estado tengan un significado físico. ¡OJO! Una de las principales virtudes de la descripción interna es realmente esta. Al trabajar con la función de transferencia, la descripción externa "sólo" aporta información sobre la relación entre la transformada de Laplace de la salida y de la

entrada del sistema. Por contra, la descripción interna nos permite trabajar con otras variables de interés. Ya veremos ejemplos de esto.

Si un sistema es lineal, las ecuaciones de estado pueden expresarse de la siguiente forma:

- Ecuación de estado: $\dot{x}(t) = Ax(t) + Bu(t)$
- Ecuación de salida: $y(t) = Cx(t) + Du(t)$

Además, si el sistema tiene n variables de estado, p entradas y m salidas, entonces:

- A , también llamada **matriz dinámica** (o de **estado**), es de dimensión $n \times n$.
- B , también llamada **matriz de control** (o de **entrada**), es de dimensión $n \times p$.
- C , también llamada **matriz de salida**, es de dimensión $m \times n$.
- D , también llamada **matriz de influencia directa**, es de dimensión $m \times p$.

Cabe destacar que cualquier estado puede representarse mediante un punto en el **espacio de estados**: espacio n -dimensional donde cada una de las n variables de estados se corresponden con un eje de coordenadas.

1.1 El comando ss

MATLAB nos facilita el trabajo con la representación interna de sistemas con el comando **ss**. Su sintaxis es:

- `sys = ss(A,B,C,D)`: Crea un modelo de espacio de estado representando el modelo continuo visto anteriormente, esto es, mediante las matrices A , B , C , y D , que definen las ecuaciones de estado y de salida. Ejemplo:

```
A = [0 1;-5 -2]; % Matriz dinámica o de estado
B = [0;3];       % Matriz de control o de entrada
C = [0 1];       % Matriz de salida
D = 0;           % Matriz de influencia directa
sys_ss = ss(A,B,C,D)
```

sys_ss =

```
A =
      x1  x2
x1      0   1
x2     -5  -2
```

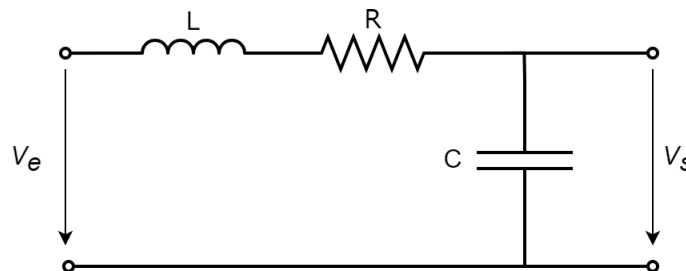
```
B =
      u1
x1      0
x2      3
```

```
C =
      x1  x2
y1      0   1
```

```
D =
      u1
y1      0
```

1.2 Ejemplo descripción interna: Circuito LRC serie

El siguiente esquema muestra un circuito LRC típico compuesto por una bobina, una resistencia y un condensador. En este circuito, la entrada al sistema es la tensión de entrada V_e , y la de salida es la caída de tensión en el condensador, V_s . Es decir, nuestro circuito cuenta con una tensión de entrada variable, y estamos interesados en controlar la caída de tensión en el condensador.



Como ingenieros de control, el primer paso para conseguirlo es modelar matemáticamente el comportamiento dinámico del sistema. Para ello se van a obtener sus ecuaciones diferenciales a partir de las leyes físicas que gobiernan el sistema. En este caso, aplicando la ley de tensiones de Kirchhoff obtenemos:

$$(1) V_e(t) = L \cdot \frac{\partial i}{\partial t} + R \cdot i(t) + \frac{1}{C} \int i \, dt$$

$$(2) V_s(t) = \frac{1}{C} \cdot \int i \, dt$$

Donde (1) expresa que la fuente de tensión del sistema $V_e(t)$ tiene que ser igual a la caída de tensión sufrida en cada uno de los componentes del mismo, y (2) expresa el valor de la caída de tensión en el condensador $V_s(t)$.

En este punto hay varias maneras de proceder en función de las variables de estado elegidas, pero en este caso perseguimos una ecuación diferencial expresada en función de la entrada y salida del sistema, por lo cual hay que buscar reemplazar la corriente $i(t)$ por una expresión en función de estas. Para ello la despejamos de (2):

$$V_s(t) = \frac{1}{C} \cdot \int i \, dt \Rightarrow C \cdot V_s(t) = \int i \, dt \Rightarrow C \cdot \frac{\partial V_s(t)}{\partial t} = i(t)$$

Sustituyendo $i(t)$ y su derivada en (1) obtenemos:

$$V_e(t) = L \cdot C \cdot \frac{\partial^2 V_s(t)}{\partial t^2} + R \cdot C \cdot \frac{\partial V_s(t)}{\partial t} + V_s(t)$$

Despejando la derivada de mayor orden (la derivada de mayor orden ha de quedar multiplicada por 1), y cambiando la notación de las derivadas por simplicidad, obtenemos:

$$\ddot{V}_s + \frac{R}{L} \dot{V}_s + \frac{1}{LC} V_s = \frac{1}{LC} V_e$$

Llegados a este punto ya podemos definir las **variables de estado**. En este caso vamos a optar por definir como primera variable de estado la salida del sistema, y como segunda su derivada. Es decir, vamos a estar interesados en la evolución y control de la caída de tensión en el condensador y de la tendencia de esta, obtenida mediante su derivada:

- $x_1 = V_s$
- $x_2 = \dot{V}_s$

Con el fin de definir la función de transición (esto es, encontrar las matrices dinámicas y de control), la cual define evolución del sistema, obtenemos las derivadas de dichas variables de estado:

- $\dot{x}_1 = \dot{V}_s = x_2$
- $\dot{x}_2 = \ddot{V}_s = \frac{1}{LC} V_e - \frac{R}{L} \dot{V}_s - \frac{1}{LC} V_s$

Y si además definimos como variable de entrada la tensión de entrada ($u = V_e$), y como variable de salida la tensión de salida ($y = V_s = x_1$), obtenemos:

- $\dot{x}_1 = x_2$
- $\dot{x}_2 = -\frac{1}{LC} x_1 - \frac{R}{L} x_2 + \frac{1}{LC} u$
- $y = x_1$

Y expresadas de manera matricial:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{LC} \end{bmatrix} \cdot u$$

\downarrow
 A

\downarrow
 B

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \cdot u$$

\downarrow
 C

\downarrow
 D

Definamos estas matrices en Matlab, fijando las constantes del circuito a $L = 0,5H$, $R = 10\Omega$, y $C = 0.001F$.

```

% Constantes del sistema
L = 0.5;
R = 10;
C= 0.001;

% Matrices del espacio de estados
A = [0 1; -1/(L*C) -R/L]; % matriz dinámica
B = [0; 1/(L*C)];         % matriz de control
C = [1 0];                 % matriz de salida
D = [0];                   % matriz de influencia directa

% Creación del modelo del sistema mediante ss
sys_LRC = ss(A,B,C,D)

```

```
sys_LRC =
```

```

A =
      x1      x2
x1      0      1
x2 -2000    -20

```

```

B =
      u1
x1      0
x2 2000

```

```

C =
      x1  x2
y1      1   0

```

```

D =
      u1
y1      0

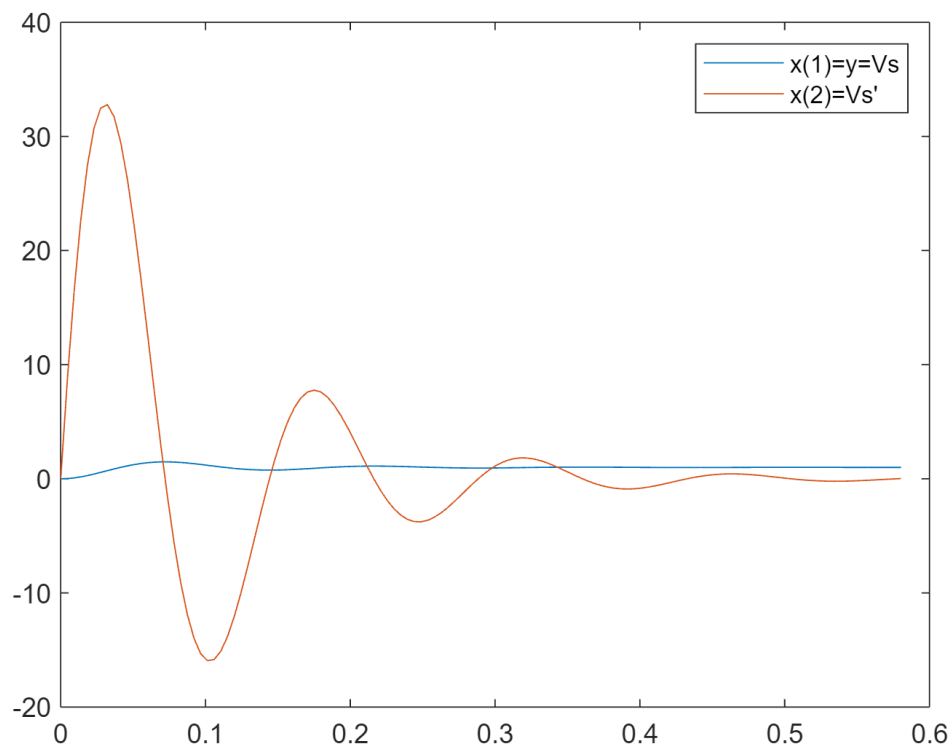
```

Continuous-time state-space model.
Model Properties

```

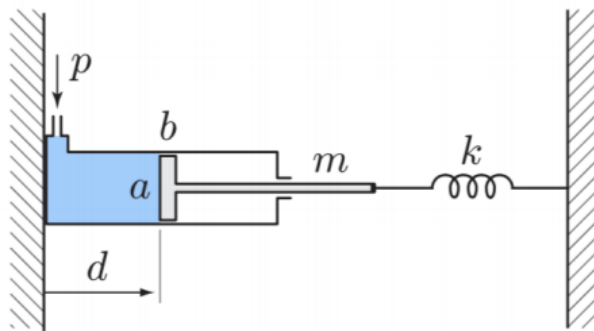
% El siguiente comando, que veremos con más detalle más adelante, simula
% la respuesta del sistema ante una entrada escalón unitario, y devuelve
% el valor de la salida y en los instantes de tiempo t, así como los
% valores de las variables de estado en x
[y,t,x] = step(sys_LRC);
% Mostramos la evolución de las variables de estado
plot(t,x), legend('x(1)=y=Vs','x(2)=Vs')

```



¡Apliquemos lo aprendido!

Tarea 1: En la figura se representa un cilindro hidráulico, cuyo émbolo de sección $a = 0,1\text{m}^2$ y masa $m = 1\text{kg}$ es accionado por una presión p de entrada.



En este sistema se mide la distancia d (salida) que se comprime un muelle de constante $k = 0,4\text{kg/s}^2$, el cual se halla en reposo con $d = 0$. Además, se halla presente una resistencia al movimiento de coeficiente $b = 0,4\text{kg/s}$, de forma que la ecuación diferencial es:

$$m\ddot{d} = -b\dot{d} - kd + ap$$

Se pide:

1. Obtener una representación interna del sistema.

% Tu código aquí

```
a = 0.1;
m = 1;
k = 0.4;
d = 0;
b = 0.4;
A = [0 1; -k/m -b/m];
B = [0;a/m];
C = [1 0];
D = [0];
sys = ss(A,B,C,D)
```

sys =

A =

	x1	x2
x1	0	1
x2	-0.4	-0.4

B =

	u1
x1	0
x2	0.1

C =

	x1	x2
y1	1	0

D =

	u1
y1	0

Continuous-time state-space model.
Model Properties

Resultado esperado:

Gss =

A =

	x1	x2
x1	0	1
x2	-0.4	-0.4

B =

	u1
x1	0
x2	0.1

C =

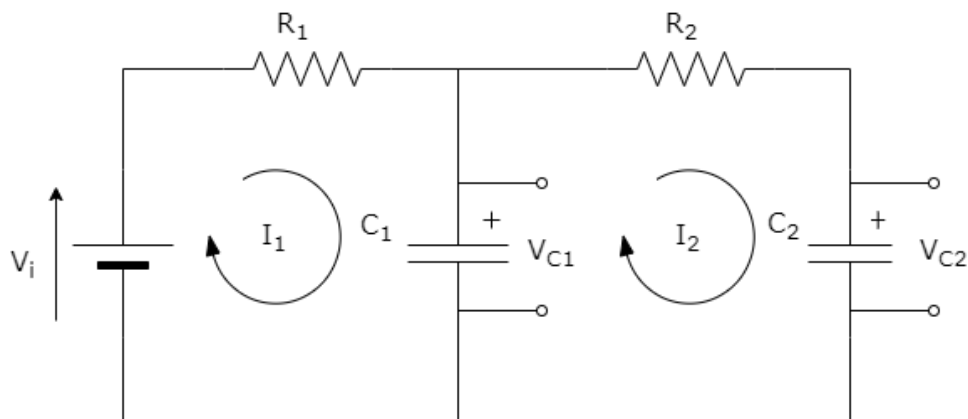
	x1	x2
y1	1	0

D =

	u1
y1	0

Continuous-time state-space model.

Tarea 2: La siguiente figura muestra el esquema eléctrico de un circuito RCRC:



El comportamiento dinámico de dicho sistema, empleando la ley de tensiones de Kirchhoff, puede modelarse matemáticamente por medio de las siguientes ecuaciones diferenciales:

$$\text{malla 1) } V_i = V_{R1} + V_{C1}$$

$$V_i = I_1 R_1 + V_{C1}$$

$$\text{malla 2) } V_{C1} = V_{R2} + V_{C2}$$

$$V_{C1} = I_2 R_2 + V_{C2}$$

Teniendo en cuenta que $V_{C1} = \frac{1}{C_1} \int (I_1 - I_2) dt$ y $V_{C2} = \frac{1}{C_2} \int I_2 dt$, obtener la representación en espacio de estados teniendo en cuenta que:

- la entrada al sistema es la tensión V_i ,
- las variables de estado serán $x_1 = V_{C1}$ y $x_2 = V_{C2}$, es decir, la caída de tensión en los condensadores, y

- la salida del sistema es la caída de tensión en el primer condensador, $y = V_{C1}$.
- Los valores que toman las resistencias y las capacitancias se dan en la celda de código.

Pista: Trata de despejar las dos corrientes y de sustituirlas en las ecuaciones obtenidas en las mallas, así se podrán obtener dos ecuaciones donde aparezcan las derivadas de las dos tensiones, facilitando la construcción de la representación en espacio de estados.

```
R1 = 8; R2 = 800; C1 = 12*10^-6; C2 = 20*10^-6;
% Tu código aquí
%Hecho en foto
A=[-1/(C1*R2)-1/(R1*C1) 1/(C1*R2); 1/(R2*C2) -1/(R2*C2)];
B=[1/(R1*C1); 0];
C=[1 0];
D=[0];

sys2 = ss(A,B,C,D)
```

sys2 =

```
A =
      x1      x2
x1 -1.052e+04  104.2
x2    62.5    -62.5
```

```
B =
      u1
x1 1.042e+04
x2    0
```

```
C =
      x1  x2
y1    1    0
```

```
D =
      u1
y1    0
```

Continuous-time state-space model.
Model Properties

Resultado esperado:

```

circuito =

A =
      x1      x2
x1 -1.052e+04  104.2
x2    62.5    -62.5

B =
      u1
x1 1.042e+04
x2      0

C =
      x1  x2
y1    1    0

D =
      u1
y1      0

Continuous-time state-space model.

```

1.3 Puntos de equilibrio

Los puntos de **equilibrio**, también llamados puntos de **operación** o **consigna**, son puntos en el espacio de estados donde el sistema puede permanecer operando con actuaciones pequeñas y constantes alrededor de dicho punto. Los puntos que cumplen con esta característica son aquellos en los que las derivadas de las variables de estado se anulan.

Estos puntos son especialmente útiles a la hora de linealizar sistemas, ya que nos permiten aproximar un sistema no lineal alrededor de dicho punto y tratarlo como lineal dentro de un cierto rango de operación. Tenemos una sesión práctica especialmente dedicada a la linealización de sistemas, pero vamos a ir adelantando trabajo con el procedimiento de obtención de **puntos de equilibrio**.

Vamos con un ejemplo. En el contexto de nuestro querido circuito LRC, recordemos cuales son las derivadas de las variables de estado:

- $\dot{x}_1 = x_2$
- $\dot{x}_2 = -\frac{1}{LC}x_1 - \frac{R}{L}x_2 + \frac{1}{LC}u$
- $y = x_1$

Si procedemos igualandolas a cero, obtenemos:

- $\dot{x}_1 = 0 \Rightarrow x_{2e} = 0$
- $\dot{x}_2 = 0 \Rightarrow -\frac{1}{LC}x_{1e} + \frac{1}{LC}u_e = 0 \Rightarrow x_{1e} = u_e$
- $y = u_e$

Estas operaciones también podemos realizarlas usando el fabuloso cálculo simbólico de MATLAB:

```
syms L C R x1 x2 ue
A = [0 1; -1/(L*C) -R/L]; % matriz dinámica
B = [0; 1/(L*C)];         % matriz de control
C = [1 0];                % matriz de salida
D = [0];                  % matriz de influencia directa
[x1e x2e] = solve(0==A*[x1;x2]+B*ue)
```

```
x1e = ue
```

```
x2e = 0
```

```
ye = C*[x1e; x2e] + D*ue
```

```
ye = ue
```

Tarea 3: Calcular los puntos de equilibrio de la descripción interna definida en la Tarea 1, tanto en papel como usando MATLAB.

Pista: Emplea `help syms` y `help solve` para obtener más información acerca de estos fantásticos comandos.

```
% Tu código aquí
```

```
syms x1 x2 ue
a = 0.1;
m = 1;
k = 0.4;
d = 0;
b = 0.4;
A=[0 1; -k/m -b/m];
B=[0; a];
C=[1 0];
D=[0];

[x1e x2e] = solve (0==A*[x1;x2]+B*ue)
```

```
x1e =
```

```
 $\frac{ue}{4}$ 
```

```
x2e = 0
```

```
ye=C*[x1e; x2e] + D*ue
```

```
ye =
```

```
 $\frac{ue}{4}$ 
```

Resultado esperado:

$$x1e =$$

$$\frac{ue}{4}$$

$$x2e = 0$$

$$ye =$$

$$\frac{ue}{4}$$