CONTROL AVANZADO: Componente Práctico

Práctica 9

Representación de sistemas en variables de estado

Representación en espacio de estados:

Un sistema se representa en espacio de estados de la siguiente manera:

$$
\begin{aligned}
\dot{x} &= Ax + Bu \\
y &= Cx + Du
\end{aligned}
$$

Donde :

$x =$ vector de estado

$\dot{x} =$ derivada de x con respecto al tiempo

$y =$ vector de salida

$u =$ vector de entrada

$A =$ matriz del sistema

$B =$ matriz de la entrada

$C =$ matriz de la salida

$D =$ matriz de prealimentacion

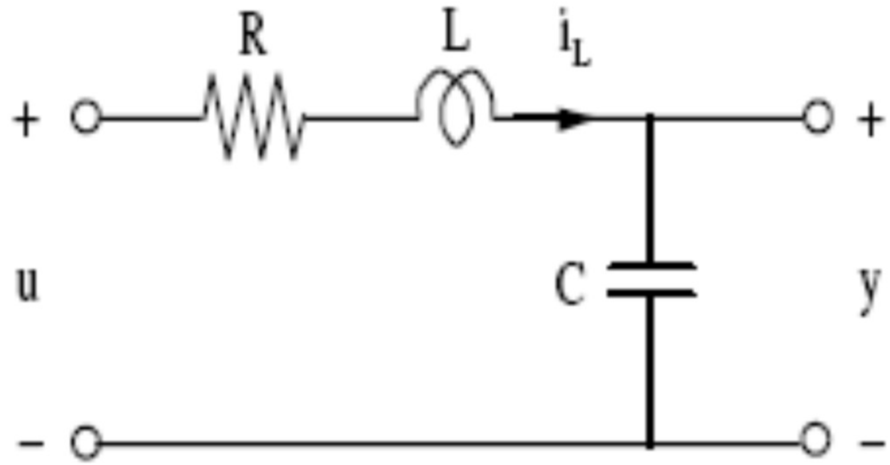
Representación en espacio de estados:

Se desarrollará lo requerido en la práctica mediante un ejemplo presentado en la clase teórica.

1) Encontrar ecuaciones diferenciales

Ejemplo:

R=5 Ω
C= 1 mF
L= 2.5 H



$$v_R = Ri_R \qquad v_L = \frac{d}{dt}(L\, i_L) = L\frac{di_L}{dt} \qquad i_C = \frac{d}{dt}(C\, v_C) = C\frac{dv_C}{dt}$$

Ley de Kirchoff: $$u = Ri_L + L\frac{di_L}{dt} + y \qquad i_L = C\frac{dy}{dt}.$$

Que se reescriben: $$L\frac{di_L}{dt} = -y - Ri_L + u \qquad C\frac{dy}{dt} = i_L.$$

Para nuestro sistema:

$$\frac{di_L}{dt} = -\frac{1}{L}y - \frac{R}{L}i_L + \frac{1}{L}u$$

$$\frac{dy}{dt} = \frac{1}{C}i_L$$

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \longrightarrow \quad A = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{R}{L} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix}.$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 \end{bmatrix}.$$

Para este circuito el voltaje del capacitor y la corriente del inductor son consideradas las variables de estado

2) Ingresar el sistema como espacio de estados en MATLAB

Comando "ss"

- Convierte a un sistema en su representación en espacio de estados.

- Sintáxis:

sys = ss(A,B,C,D)

- Argumentos:

  - ❑ A — Matriz de estados
  - ❑ B — Matriz de entrada a estado
  - ❑ C — Matriz de estado a salida
  - ❑ D — Matriz de alimentación

Comando "ss"

```
R=5;
L=2.5;
C=1e-3;

% 1) Espacio de estados

A=[0 1/C; -1/L -R/L];
B=[0;1/L];
C=[1 0];
D=0;

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

```
Gss =

 A =
        x1    x2
 x1      0  1000
 x2   -0.4    -2

 B =
       u1
 x1     0
 x2   0.4

 C =
       x1  x2
 y1     1   0

 D =
       u1
 y1     0
```

3) Encontrar la FT analíticamente

A partir del sistema físico y usando circuito transformado podemos obtener:

$$U(s) = (R + sL + \frac{1}{sC})I(s); \qquad Y(s) = \frac{1}{sC}I(s)$$

$$\frac{Y(s}{U(s)} = \frac{1}{s^2LC + sRC + 1} = \frac{\frac{1}{LC}}{s^2 + \frac{R}{L}s + \frac{1}{LC}}$$

De donde reemplazando los valores numéricos obtenemos:

$$\frac{Y(s)}{U(s)} = \frac{400}{s^2 + 2s + 400}$$

4) Comprobar en MATLAB la estimación de la FT

Comando "tf"

- Si se aplica sobre una función descrita en espacio de estados, nos entrega la matriz función de transferencia del sistema:

$$Gs=tf(Gss)$$

```
Gs =

      400
  ---------------
  s^2 + 2 s + 400

Continuous-time transfer function.
```

5) Obtener la FT teórica para cualquier representación en EE

- La matriz función de transferencia de un sistema se obtiene con la siguiente fórmula:

$$T(s) = Y(s).U(s)^{-1} = C(sI - A)^{-1}B + D$$

6) Obtener FCC y FCO en forma analítica a partir de la FT

- Partiendo de la función de transferencia obtenida:

$$\frac{Y(s)}{U(s)} = \frac{400}{s^2 + 2s + 400}$$

- Se pueden estimar las matrices de las FCC y FCO:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -400 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 400 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & -400 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 400 \\ 0 \end{bmatrix} u$$

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

7) Obtener FCC y FCO en MATLAB

Comando “canon”

- Transforma un sistema lineal (sys) a su forma canónica en espacio de estados.

$$\mathrm{csys} = \mathrm{canon}(\mathrm{sys,type})$$

- “type” especifica si “csys” está en la foma compañía (companion) o en forma modal (una versión de la forma diagonal).

- Las formas canónicas controlables y observables corresponden al tipo “companion”

$$\mathrm{csys} = \mathrm{canon}(\mathrm{sys,'companion'})$$

Comando “canon”

- La aplicación del comando retorna la información de las matrices pero no en la exacta forma esperada.

Gcan=canon(Gs,'companion')

- Es una versión de la forma canónica observable.

- Se tienen que hacer ajustes para convertir las matrices a las formas FCC y FCO tradicionales.

Gcan =

A =

	x1	x2
x1	0	-400
x2	1	-2

B =

	u1
x1	1
x2	0

C =

	x1	x2
y1	0	400

D =

	u1
y1	0

Forma Canónica Controlable

```
Acc=Gcan.A'; Bcc=flip(Gcan.B);
Ccc=flip(Gcan.C);Dcc=Gcan.D;

Gcc=ss(Acc,Bcc,Ccc,Dcc)
```

```
Gcc =

 A =
        x1    x2
 x1      0     1
 x2   -400    -2

 B =
      u1
 x1    0
 x2    1

 C =
        x1   x2
 y1    400    0

 D =
      u1
 y1    0
```

Forma Canónica Observable

```
Aco=Gcan.A; Bco=flip(Gcan.C');
Cco=flip(Gcan.B');Dco=Gcan.D;

Gco=ss(Aco,Bco,Cco,Dco)
```

```
Gco =

 A =
        x1    x2
 x1     0  -400
 x2     1    -2

 B =
        u1
 x1    400
 x2      0

 C =
       x1  x2
 y1     0   1

 D =
       u1
 y1     0
```