

Práctica 4

Modelos Matemáticos de Sistemas Lineales

Objetivo. El objetivo de esta práctica es dar una introducción al uso de algunas funciones de Matlab orientadas al manejo de modelos matemáticos de sistemas lineales en función de transferencia, tales como `tf`, `zpk`, `zpk2tf`, `tf2zpk` y `residue`.

Introducción.

El toolbox de sistemas de control (Control Systems Toolbox) de Matlab proporciona un conjunto de funciones que van desde las funciones básicas para introducir modelos de sistemas lineales invariantes en el tiempo hasta herramientas interactivas para el análisis y diseño de sistemas de control.

Un sistema lineal invariante en el tiempo (SLIT por sus siglas en español y LTI por sus siglas en inglés) es aquel que cumple con el principio de superposición. Si el sistema tiene una entrada $u(t)$ y una salida $y(t)$, su representación en el dominio del tiempo corresponde a una ecuación diferencial lineal con coeficientes constantes como la siguiente:

$$a_n^{(n)} y + a_{n-1}^{(n-1)} \dot{y} + \dots + a_1 \ddot{y} + a_0 y = b_m^{(m)} u + b_{m-1}^{(m-1)} \dot{u} + \dots + b_1 \ddot{u} + b_0 u \quad (4.1)$$

donde los coeficientes $a_0, a_1, \dots, a_n, b_0, b_m, \dots, b_m$ son constantes, y donde la notación $y^{(k)}$ representa la k-ésima derivada $\frac{d^k y}{dt^k}$, $k=0,1,2,\dots,n$.

Aplicando Transformada de Laplace a ambos miembros de la ecuación (4.1), y suponiendo condiciones iniciales cero, podemos despejar el cociente $\frac{Y(s)}{U(s)}$, para obtener la denominada Función de Transferencia del sistema:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \quad (4.2)$$

Obsérvese que la función de transferencia de un sistema del tipo (4.1) siempre es una función racional, es decir, una división de polinomios en potencias de la variable de Laplace (s).

Una de las virtudes de la representación en función de transferencia (4.2), es que permite escribir la salida $Y(s)$ como una simple multiplicación de la entrada $U(s)$ por $G(s)$, es decir,

$$Y(s) = G(s)U(s) \quad (4.3)$$

En otras palabras, la función de transferencia $G(s)$ viene siendo la **ganancia** aplicada a la entrada $U(s)$ para obtener la salida $Y(s)$.

El razonamiento anterior sustentado en la expresión (4.3) permite obtener la regla más simple de la representación en diagrama de bloques del sistema dado por la ecuación diferencial (4.1) de acuerdo a la figura siguiente

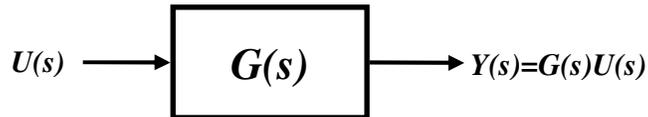


Figura 4.1.- Diagrama de Bloques del sistema dado por (4.1)

Ejemplo: El sistema dado por la ecuación diferencial $2\ddot{y} + 6\dot{y} + 4y = \dot{u} + 3u$ tiene la representación en función de transferencia siguiente

$$G(s) = \frac{s+3}{2s^2+6s+4} \quad (4.4)$$

La representación (4.4) no es única, ya que numerador y denominador son polinomios en potencias de s , se pueden escribir de muchas maneras. Algunas de ellas son las siguientes:

Forma factorizada: Todo polinomio se puede factorizar en binomios de primer orden. Esta forma se prefiere cuando se quieren evidenciar los **polos** y los **ceros** de la función de transferencia, así, por ejemplo, la función de transferencia (4.4) se puede escribir en su forma factorizada como sigue

$$G(s) = \frac{s+3}{(2s+4)(s+1)}$$

y procurando que todos los factores de primer orden tengan coeficiente 1 en la variable s , se obtiene la forma estándar:

$$G(s) = \frac{0.5(s+3)}{(s+2)(s+1)} \quad (4.5)$$

De la forma (4.5) resultan evidentes:

los **polos**: -2 y -1,

los **ceros**: -3

la **ganancia** del sistema: 0.5

Forma de fracciones parciales: Usando la técnica de expansión en fracciones parciales, la forma (4.4) también puede expandirse en una suma de fracciones, cada una con denominador de primer orden (llamadas fracciones simples o parciales) una por cada polo. Para el ejemplo anterior, la forma de fracciones parciales queda

$$G(s) = \frac{1}{s+1} - \frac{0.5}{s+2} \quad (4.6)$$

La forma de fracciones parciales es de interés en el problema de diseño, ya que permite una implementación de la función de transferencia como la conexión en paralelo de subsistemas sencillos de primer orden.

Comandos de Matlab.

Matlab proporciona comandos para introducir funciones de transferencia en las diferentes formas descritas arriba. Estos comandos se describen a continuación.

El comando `tf`.

Es el comando básico para introducir funciones de transferencia (F. T.) de sistemas SISO (una entrada - una salida) o bien, matrices de funciones de transferencia de sistemas MIMO (múltiples entradas - múltiples salidas).

Ejemplo: Se puede introducir la función de transferencia dada por (4.4) de dos maneras:

1) Definiendo los polinomios del numerador y denominador por sus coeficientes

```
>> num=[1 3];      %%Define numerador
>> den=[2 6 4];   %%Define denominador
>> G=tf(num,den)  %%Define la F.T.
Transfer function:
      s + 3
-----
2 s^2 + 6 s + 4
```

2) Definiendo primero la variable de Laplace y luego escribiendo la expresión algebraica (4.4)

```
>> s=tf('s');      %%Define la variable de Laplace
>> G=(s+3)/(2*s^2+6*s+4)%%Define la F.T.
Transfer function:
      s + 3
-----
2 s^2 + 6 s + 4
```

El comando `zpk`.

Este comando también permite introducir una función de transferencia a partir de sus polos, sus ceros y su ganancia, es decir, cuando se tiene escrita en la forma factorizada.

Ejemplo: Para introducir la F.T. dada por (4.5) se procede como sigue:

```
>> K=0.5;          %%Ganancia del sistema
>> ceros=[-3];    %%Vector de ceros
>> polos=[-2 -1]; %%Vector de polos
>> G=zpk(ceros,polos,K)%%Define la F.T.

Zero/pole/gain:
 0.5 (s+3)
-----
(s+2) (s+1)
```

El comando `residue`.

Este comando como ya fue explicado en la práctica No. 2. permite obtener la representación de

una división de polinomios en suma de fracciones parciales.

Ejemplo: Para obtener la representación de la F. T. (4.4) en su forma de fracciones parciales se procede como sigue:

```
>> num=[1 3];
>> den=[2 6 4];
>> [R,P,K]=residue(num,den)
R =
    -0.5000
     1.0000
P =
    -2
    -1
K =
     []
```

Es decir, la representación de (4.4) en fracciones parciales es la esperada (4.6), es decir, los numeradores de las fracciones están dados por el vector R, los polos de los denominadores están dados por P y el término directo (sin denominador) está dado por K (que en este caso es vacío).

Los comandos `tf2zp` y `zp2tf`.

Estos comandos permiten realizar la conversión entre las dos representaciones básicas de una F.T., es decir, de la forma de división de polinomios en potencias descendentes de s a la forma factorizada y viceversa.

Ejemplo: Para convertir la F.T: de la forma (4.4) a la forma (4.5) se procede como sigue

```
>> num=[1 3];
>> den=[2 6 4];
>> [Z,P,K]=tf2zp(num,den)
Z =
    -3
P =
    -2
    -1
K =
     0.5000
>> G=zpk(Z,P,K)
```

```
Zero/pole/gain:
 0.5 (s+3)
-----
(s+2) (s+1)
```

Ejemplo: Para convertir la F.T: de la forma (4.5) a la forma (4.4) se procede como sigue

```
>> Z=[-3]; %%los corchetes son opcionales pues es solo un cero
>> P=[-2 -1];%%define polos
>> K=0.5; %%ganancia
>> [num,den]=zp2tf(Z,P,K)%%Convierte
num =
     0     0.5000     1.5000
den =
     1     3     2
>> G=tf(num,den)
Transfer function:
```

$$\frac{0.5 s + 1.5}{s^2 + 3 s + 2}$$

* Obsérvese que el resultado es una variante de (4.4) con el coeficiente de la máxima potencia del denominador igual a 1.

Los comandos presentados arriba se pueden complementar con los comandos vistos para manejar polinomios explicados en la práctica No. 2. A continuación se dan algunos otros ejemplos:

Ejemplo: En ocasiones la F.T. no está escrita en ninguna de las tres formas mencionadas arriba,

sino en una combinación de ellas, por ejemplo: $G(s) = \frac{s^2 + 2}{s(s^2 + 3s + 2)(s + 0.5)}$. En este caso se

puede introducir la F.T. de diferentes maneras:

1) Usando multiplicación de polinomios mediante el comando `conv`.

```
>> num=[1 0 2]; %Numerador
>> d1=[1 0]; %factor s
>> d2=[1 3 2]; %factor s^2+3s+2
>> d3=[1 0.5]; %factor s+0.5
>> den=conv(d1,d2);
>> den=conv(den,d3);%Obtiene denominador
>> G=tf(num,den) %define F.T.
Transfer function:
      s^2 + 2
-----
s^4 + 3.5 s^3 + 3.5 s^2 + s
```

2) Factorizando en tres F.T. y luego multiplicándolas.

```
>> G1=tf([1 0 2], [1 3 2])
Transfer function:
      s^2 + 2
-----
s^2 + 3 s + 2
>> G2=tf(1, [1 0])
Transfer function:
      1
-----
s
>> G3=tf(1, [1 0.5])
Transfer function:
      1
-----
s + 0.5
>> G=G1*G2*G3 %Multiplica los tres factores
Transfer function:
      s^2 + 2
-----
s^4 + 3.5 s^3 + 3.5 s^2 + s
```

3) Definiendo la variable de Laplace y escribiendo la expresión algebraica correspondiente a la F.T.

```
>> s=tf('s');
>> G=(s^2+1)/s/(s^2+3*s+2)/(s+0.5)
```

Transfer function:

$$\frac{1}{s^2 + 1}$$

$$\frac{1}{s^4 + 3.5s^3 + 3.5s^2 + s}$$

Caso MIMO:

Cuando un sistema tiene varias entradas y varias salidas, la noción de Función de Transferencia se puede generalizar, a la de una matriz que contiene como elementos funciones de transferencia escalares y que se denomina Matriz de Transferencia. En la figura 4.2 se muestra en forma de bloque un sistema con p entradas y q salidas.

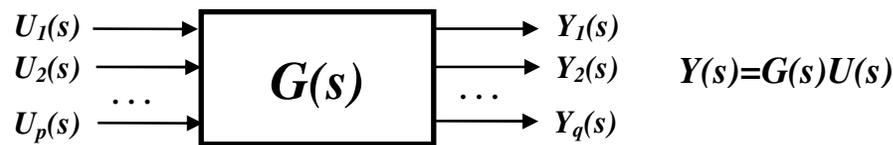


Figura 4.2.- Diagrama de Bloques de un sistema con p entradas y q salidas

Para el sistema MIMO de la figura 4.2 sigue siendo válida la relación

$$Y(s) = G(s)U(s) \quad (4.7)$$

sin embargo, (4.7) es ahora una expresión matricial donde:

$$Y(s) = \begin{bmatrix} Y_1(s) \\ Y_2(s) \\ \dots \\ Y_q(s) \end{bmatrix}, \quad U(s) = \begin{bmatrix} U_1(s) \\ U_2(s) \\ \dots \\ U_p(s) \end{bmatrix}, \quad G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \dots & G_{1p}(s) \\ G_{21}(s) & G_{22}(s) & \dots & G_{2p}(s) \\ \vdots & \vdots & \ddots & \vdots \\ G_{q1}(s) & G_{q2}(s) & \dots & G_{qp}(s) \end{bmatrix} \quad (4.8)$$

Donde $G(s)$ es la matriz de transferencia cuyos elementos se pueden obtener por superposición como sigue:

$$G_{ij}(s) = \left. \frac{Y_i(s)}{U_j(s)} \right|_{U_{k \neq j}(s)=0} \quad \text{para } i = 1, 2, \dots, q, \quad j = 1, 2, \dots, p \quad (4.9)$$

Tanto el comando `ft` como el comando `zpk` permiten introducir matrices de transferencia en Matlab.

Ejemplo: Para introducir en Matlab la Matriz de Transferencia: $G(s) = \begin{bmatrix} \frac{1}{s^2 + s + 1} & \frac{s}{s + 1} \\ \frac{s + 2}{s + 1} & \frac{s}{s^2 + s + 1} \end{bmatrix}$ se

puede proceder como sigue:

```
>> num11=1; num12=[1 0];
>> num21=[1 2]; num22=[1 0];
>> den11=[1 1 1]; den12=[1 1];
>> den21=[1 1]; den22=[1 1 1];
>> help tf
>> G=tf({num11,num12;num21,num22},{den11,den12;den21,den22})
```

O también:

```
>> num11=1; num12=[1 0];
>> num21=[1 2]; num22=[1 0];
>> G11= tf(num11,den11); G12= tf(num12,den12);
>> G21= tf(num21,den21); G22=tf(num22,den22);
>> G=[G11,G12;G21,G22]
```

O también, usando la variable de Laplace:

```
>> s=tf('s');
>> G=[1/(s^2+s+1), s/(s+1);(s+2)/(s+1), s/(s^2+s+1)]
```

Una vez que se ha introducido correctamente la matriz de transferencia $G(s)$ por cualquiera de los tres métodos anteriores, Matlab la desplegará por columnas:

Transfer function from input 1 to output...

```
      1
#1:  -----
      s^2 + s + 1
      s + 2
#2:  -----
      s + 1
```

Transfer function from input 2 to output...

```
      s
#1:  -----
      s + 1
      s
#2:  -----
      s^2 + s + 1
```

Ejercicio: Escribir una función de Matlab en un archivo-M que a partir de los polinomios del numerador y denominador de una F. T. obtenga y despliegue las tres formas básicas descritas arriba (En potencias descendentes, en forma factorizada y en suma de fracciones parciales). El formato para ejecutar la función deberá ser: $G=\text{fun_transf}(\text{num}, \text{den})$.

Desarrollo de la Práctica.

1. Probar todos los ejemplos propuestos por el profesor conforme los va explicando.
2. Realizar todos los ejercicios propuestos.
3. Contestar el cuestionario de evaluación de la práctica.

Reportar:

1) El código escrito y comentado para la función `fun_transf` del ejercicio propuesto acompañado de dos ejemplos de ejecución.

2) Obtener la F.T. $\frac{V_C(s)}{V_i(s)}$ correspondiente al siguiente circuito y escribir una función que la defina

en términos de los valores de R,L y C en Matlab, de la forma $G=FT_VcVi(R,L,C)$.

