Práctica 4
Modelos Matemáticos de Sistemas Lineales

Objetivo. El objetivo de esta práctica es dar una introducción al uso de algunas funciones de Scilab orientadas al manejo de modelos matemáticos de sistemas lineales invariantes en el tiempo, en función de transferencia y en variables de estado, tales como **`syslin`**, **`poly`**, **`pfss`**, **`trfmod`**, **`tf2ss`**, **`ss2tf`** y **`abcd`**.

Introducción.

El toolbox CACSD (Computer Aided Control Systems Design) de Scilab proporciona un enorme conjunto de funciones que van desde las funciones básicas para introducir modelos de sistemas lineales invariantes en el tiempo (SLIT's) hasta herramientas poderosas para el análisis y diseño de sistemas de control tanto de una entrada - una salida (SISO) o de múltiples entradas y múltiples salidas (MIMO).

El modelo en Función de Transferencia.

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:

$$a_n \overset{(n)}{y} + a_{n-1} \overset{(n-1)}{y} + \ldots + a_1 \dot{y} + a_0 y = b_m \overset{(m)}{u} + b_{m-1} \overset{(m-1)}{u} + \ldots + b_1 \dot{u} + b_0 u \qquad (4.1)$$

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

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

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 $Y(s)$ del sistema como una simple multiplicación de la entrada $U(s)$ por $G(s)$, es decir,

$$Y(s) = G(s)U(s) \tag{4.3}$$

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

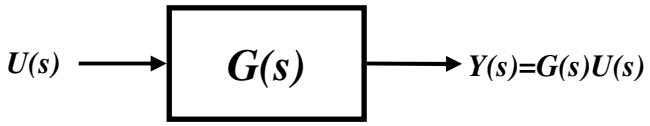

Figura 4.1.- Diagrama de Bloques del sistema dado por (4.1)

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

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

La representación (4.4) no es única, ya que numerador y denominador son polinomios en potencias de s y 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 <u>todos</u> los factores de primer orden tengan coeficiente 1 en la variable s, se obtiene la forma estándar:

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

De la forma (4.5) resultan evidentes:

los ***polos***: -2 y -1,
los ***ceros***: -3
la ***ganancia*** del sistema: 0.5

☞ **Observación**: Aunque todo polinomio se puede factorizar como producto de binomios de primer orden, cuando el polinomio tiene raíces complejas esta factorización conduce al uso de números complejos, por esta razón se prefiere dejar <u>sin factorizar</u> los factores cuadráticos que tienen raíces complejas conjugadas.

Ejemplo 4.1. El polinomio $p(s)=s^3+3s^2+4s+4$ se puede factorizar como el producto de tres binomios de primer orden, un binomio por cada una de sus raíces, reales o complejas, es decir, $p(s)=(s+2)(s+0.5+\frac{\sqrt{7}}{2}i)(s+0.5-\frac{\sqrt{7}}{2}i)$, pero para evitar el uso de números complejos se preferirá factorizar como $p(s)=(s+2)(s^2+s+2)$. Es decir, los dos binomios con raíces complejas se convierten en el trinomio cuadrático (s^2+s+2). Mediante comandos de Scilab esto se puede verificar como sigue

```
p=poly([ 4 4 3 1],'s','c')//define el polinomio mediante sus coeficientes
p  =
                2   3
    4 + 4s + 3s + s
roots(p) //calcula sus raíces
ans  =
  - 2.
  - 0.5 + 1.3228757i
  - 0.5 - 1.3228757i
polfact(p) //obtiene los factores irreducibles sin usar números complejos
ans  =
                              2
    1     2 + s     2 + s + s
```

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

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

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.

El comando `syslin` para definir funciones de transferencia (F.T.)

Es el comando básico para introducir modelos de sistemas lineales tanto en función de transferencia (F.T.) como en variables de estado, para sistemas SISO (una entrada - una salida) y para sistemas MIMO (múltiples entradas - múltiples salidas), es el comando **`syslin`**. Primero se describirá su uso para la introducción de modelos en función de transferencia.

La sintaxis del comando `syslin` es **`G=syslin(dominio,num,den)`**. Este comando define la función de transferencia **`G`** con el numerador especificado por el polinomio **`num`** y el denominador especificados por el polinomio **`den`**. El **`dominio`** puede ser '**`c`**' para un sistema continuo (analógico) o '**`d`**' para un sistema discret (digital).

Ejemplo 4.2. Se puede introducir la función de transferencia dada por (4.4) de tres maneras:

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

```
num=poly([3 1],'s','c');//Define numerador a partir de sus coeficientes
den=poly([4 6 2],'s','c');//Define denomin. a partir de sus coeficientes
G=syslin('c',num,den)   //Define la F.T. continua
G  =
        3 + s
    -----------
                2
    4 + 6s + 2s
```

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

```
s=poly(0,'s');    //Define la variable de Laplace
G=(s+3)/(2*s^2+6*s+4)   //Define la F.T.
G  =
        3 + s
    -----------
                2
    4 + 6s + 2s
```

3) Utilizando la forma factorizada dada por (4.5) se procede como sigue:

```
K=0.5;                  //Ganancia del sistema
ceros=poly([-3],'s');   //Numerador definido por su vector de raíces
polos=poly([-2 -1],'s');//Denominador definido por su vector de raíces
G=syslin('c',K*ceros,polos)   //Define la F.T.
G  =
    1.5 + 0.5s
    ----------
                2
    2 + 3s + s
```

El comando `trfmod` y la forma factorizada de una F.T.

El comando **`trfmod`** permite visualizar en una ventana los factores irreducibles del numerador y denominador de una función de transferencia de un sistema SISO.

Ejemplo 4.3: Para visualizar la misma F.T. dada del ejemplo anterior una vez definida por cualquiera de los tres métodos descritos, se ejecuta el comando `trfmod` como sigue y Scilab despliega la ventana mostrada en la figura 4.2, en la cual se pueden ver claramente: la ganancia y los factores del numerador y del denominador, los cuales coinciden con la forma factorizada (4.5)

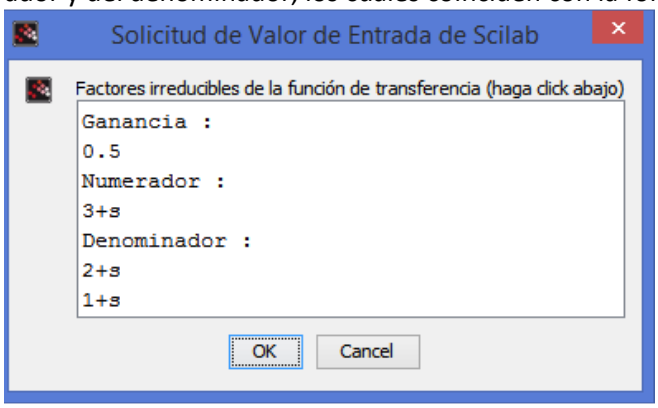

Figura 4.2. Ventana desplegada por el comando `trfmod(G,'p')`

```
trfmod(G,'p'); //el parámetro 'p' es opcional
```

El comando `pfss` y la forma de fracciones parciales.

El comando **`pfss(G)`** realiza el cálculo de la expansión en fracciones parciales de la función de transferencia especificada G.

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

```
num=poly([1 3],'s','c');// Define numerador
den=poly([2 6 4],'s','c');// define denominador
G=syslin('c',num,den); //Define F.T.
frac=pfss(G) //Calcula vector de fracciones parciales
frac  =
 frac(1)
      1
    -----
    1 + s
 frac(2)
    - 0.5
    -----
    2 + s
```

Es decir, la representación de (4.4) en fracciones parciales es la esperada (4.6). Obsérvese que la suma de estas fracciones parciales produce la F.T. original como es de esperarse:

```
frac(1)+frac(2) //Recalcula la función original.
ans  =
    1.5 + 0.5s
    ----------
              2
    2 + 3s + s
```

Los comandos presentados arriba se pueden complementar con los comandos vistos para manejar polinomios explicados en la práctica No. 2. A continuación se da un ejemplo.

Ejemplo 4.5: En ocasiones la F.T. no está escrita en ninguna de las tres formas mencionadas arriba, sino en una combinación de ellas, por ejemplo: $G(s)=\dfrac{s^2+2}{s(s^2+3s+2)(s+0.5)}$. En este caso se puede introducir la F.T. de diferentes maneras:

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

```
num=poly([2 0 1],'s','c'); //Numerador
d1=[0 1];    //coeficientes del factor s
d2=[2 3 1];  //coeficientes del factor s^2+3s+2
d3=[0.5 1];  //coeficientes del factor s+0.5
den=conv(d1,d2);
den=conv(den,d3);       //Obtiene coeficientes del denominador
den=poly(den,'s','c');  //convierte coeficientes a polinomio
G=syslin('c',num,den)   //define F.T.
G  =
                2
```

```
          2 + s
    ------------------
            2      3   4
    s + 3.5s + 3.5s + s
```

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

```
num1=poly([2 0 1],'s','c');
den1= poly([2 3 1],'s','c');
G1=syslin('c',num1,den1)
G1  =
          2
      2 + s
    ---------
              2
    2 + 3s + s
num2=poly([1],'s','c');
den2= poly([0 1],'s','c');
G2=syslin('c',num2,den2)
G2  =
    1
    -
    s
num3=poly([1],'s','c');
den3= poly([0.5 1],'s','c');
G3=syslin('c',num3,den3)
G3  =

       1
    -------
    0.5 + s
G=G1*G2*G3  //Multiplica los tres factores
G  =
                2
          2 + s
    ------------------
            2      3   4
    s + 3.5s + 3.5s + s
```

3) Definiendo la variable de Laplace 's' y escribiendo la expresión algebraica de la F.T.

```
s=poly([0],'s'); //los corchetes del cero son opcionales
G=(s^2+2)/s/(s^2+3*s+2)/(s+0.5)
G  =
                2
          2 + s
    ------------------
            2      3   4
    s + 3.5s + 3.5s + s
```

Ejemplo de obtención de un modelo en función de transferencia.

Consideremos el circuito RLC serie de la figura 4.3a Para obtener un modelo en función de transferencia lo primero que se debe saber es cuales son las variables de **entrada** y de **salida**. En este caso se desea obtener la función de transferencia que relaciona el voltaje v_c en el capacitor

(salida) al voltaje V_i aplicado por la fuente (entrada).

La manera más sencilla de obtener el modelo en función de transferencia de un sistema es expresar sus componentes directamente en el dominio de Laplace, es decir, en forma de ***impedancias complejas***, como se muestra en la figura 4.3b

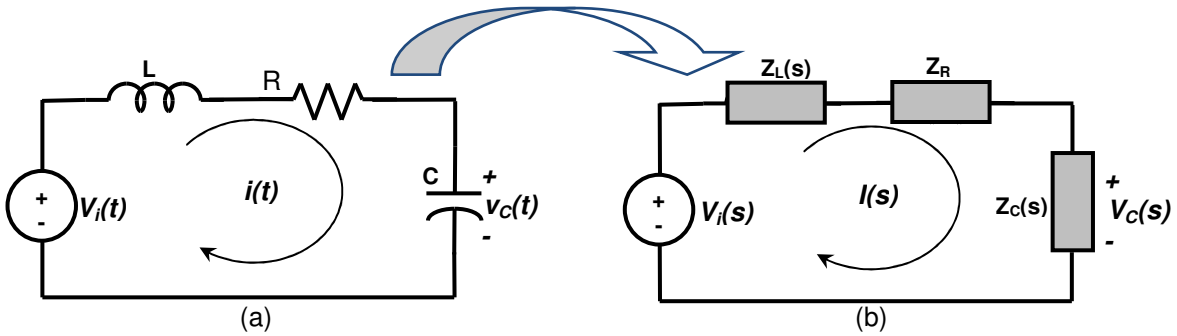

Figura 4.3 a) Circuito RLC serie. b) Circuito RLC serie en términos de ***impedancias complejas***.

El modelo de impedancias complejas permite aplicar la ley de Ohm directamente sobre cada impedancia para obtener la caída de voltaje entre las terminales de cada componente directamente en el dominio de Lapalace de acuerdo a la siguiente Tabla

Componente	**Relación v(t), i(t)**	**Relación V(s), I(s)**	**Impedancia Z=V(s)/I(s)**
Resistencia (R)	$v(t) = R\,i(t)$	$V(s) = R\,I(s)$	$Z_R = R$
Inductancia (L)	$v(t) = L\dfrac{di(t)}{dt}$	$V(s) = Ls\,I(s)$	$Z_L = Ls$
Capacitancia (C)	$v(t) = \dfrac{1}{C}\int_0^t i(t)dt$	$V(s) = \dfrac{1}{Cs}I(s)$	$Z_C = \dfrac{1}{Cs}$

Por lo tanto, aplicando la regla del divisor de tensión al circuito de la figura 4.3b obtenemos

$$V_c(s) = \frac{Z_C}{Z_L + Z_R + Z_C}V_i(s) \tag{4.7}$$

y sustituyendo las expresiones de la tabla anterior se obtiene

$$V_c(s) = \frac{\dfrac{1}{Cs}}{sL + R + \dfrac{1}{Cs}}V_i(s)$$

Simplificando

$$V_c(s) = \frac{1}{LCs^2 + RcS + 1}V_i(s)$$

Por lo tanto, la función de transferencia buscada es

$$G(s) = \frac{V_c(s)}{V_i(s)} = \frac{1}{LCs^2 + RcS + 1} \tag{4.8}$$

Modelo en Variables de Estado (o de Espacio de Estado).

Consideremos un SLIT SISO cuya entrada es $u(t)$ y salida $y(t)$, el cual se puede representar por la ecuación diferencial lineal de orden n dada por (4.1), la cual se reescribe a continuación

$$a_n \overset{(n)}{y} + a_{n-1} \overset{(n-1)}{y} + \ldots + a_1 \dot{y} + a_0 y = b_m \overset{(m)}{u} + b_{m-1} \overset{(m-1)}{u} + \ldots + b_1 \dot{u} + b_0 u \tag{4.9}$$

Aunque no siempre es sencillo, se puede elegir un conjunto de ***variables de estado*** (que en general pueden ser sumas ponderadas de la entrada, la salida y/o algunas de sus derivadas) para transformar (4.9) en ***n ecuaciones diferenciales lineales de primer orden*** llamadas ***ecuaciones de estado***, de la forma

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + b_1 u \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + b_2 u \\ &\ldots \\ \dot{x}_n &= a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n + b_n u \end{aligned} \tag{4.10}$$

Además, la salida del sistema normalmente es una combinación de estados y de la entrada que se puede escribir como la ***ecuación de salida*** siguiente

$$y = c_1 x_1 + c_2 x_2 + \ldots . c_n x_n + du \tag{4.11}$$

Es decir, en forma matricial compacta

$$\dot{x} = Ax + bu \tag{4.12}$$

$$y = cx + du \tag{4.13}$$

Donde x es el vector de estados, A es una matriz cuadrada $n \times n$ de coeficientes constantes, b es un vector columna de coeficientes constantes, c es un vector renglón de coeficientes constantes y d es un escalar constante (en la mayoría de los casos $d = 0$), es decir,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},\ \dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix},\ A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix},\ b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},\ c = \begin{bmatrix} c_1 & c_2 & \ldots & c_n \end{bmatrix} \tag{4.14}$$

Las ***ecuaciones de estado*** (4.12) y la ***ecuación de salida*** (4.13) se pueden transformar de manera directa a su Función de Transferencia equivalente aplicando Transformada de Laplace suponiendo condiciones iniciales cero, obteniéndose

$$\begin{aligned} sX(s) &= AX(s) + bU(s) \\ Y(s) &= cX(s) + dU(s) \end{aligned} \tag{4.15}$$

despejando el vector de estado

$$X(s) = (sI - A)^{-1} bU(s) \tag{4.16}$$

sustituyendo en la ecuación de salida

$$Y(s)=\left[c(sI-A)^{-1}b+d\right]U(s) \tag{4.17}$$

Es decir, la función de transferencia correspondiente al modelo en variables de estado (4.12), (4.13) es

$$G(s)=\frac{Y(s)}{U(s)}=c(sI-A)^{-1}b+d \tag{4.18}$$

☞ **Observación**: Dado un modelo en variables de estado, le corresponde una única función de transferencia dada por (4.18), sin embargo, dada una función de transferencia, ésta puede corresponder a una infinidad de modelos en variables de estado válidos, dependiendo del conjunto de variables de estado elegido. Esta situación se ilustra en la figura 4.3

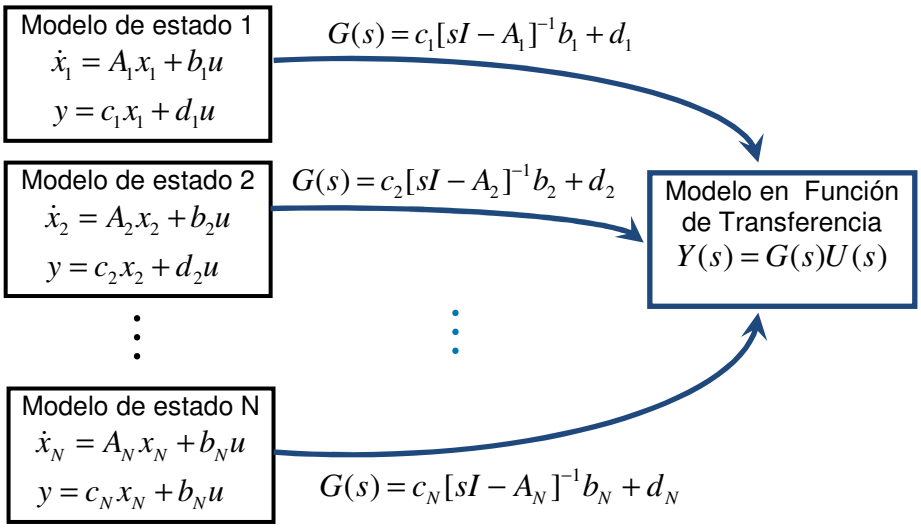

Figura 4.3. A una sola F. T. le corresponde una infinidad de Modelos en Variables de Estado

Comandos de Scilab para manejar modelos en Espacio de Estados.

Scilab provee los comandos explicados en la siguiente tabla para manejar modelos en el espacio de estado de SLIT's continuos.

Comando	**explicación**
`sist=syslin('c',A,B,C,D,x0)`	Crea un sistema tipo `SS` (Espacio de Estado), es decir, `sist` contiene la información del modelo de espacio de estado: dx/dt = Ax(t) + Bu(t) y(t) = Cx(t) + Du(t) con las condiciones iniciales x(0)=`x0` (opcionales)
`[A,B,C,D]=abcd(sist)`	Obtiene las matrices A,B,C,D correspondientes al modelo de espacio de estado de `sist`.
`G=ss2tf(sist)`	Convierte el modelo de espacio de estado dado por `sist` a la matriz de transferencia correspondiente G. Es decir, evalúa la ecuación (4.18) para el caso MIMO.
`sist=tf2ss(G)`	Convierte la matriz de funciones de transferencia G a una de sus representaciones en espacio de estado.

Ejemplo 4.6. Conversión entre F.T. y Espacio de Estado.

```
num=poly([1 1],'s','c');
den=poly([6 5 1],'s','c');
G=syslin('c',num,den) // Define la F.T. G=(s+1)/(s^2+5*s+6)
G  =
      1 + s
    ---------
              2
    6 + 5s + s

sist=tf2ss(G)// Obtiene modelo de espacio de estado correspondiente a G
 sist  =
       sist(1)   (state-space system:)
!lss  A  B  C  D  X0  dt  !
       sist(2) = A matrix =
  - 4.4  - 0.8
    4.2  - 0.6
       sist(3) = B matrix =
  - 1.2649111
    0.6324555
       sist(4) = C matrix =
  - 0.7905694    5.551D-17
       sist(5) = D matrix =
    0.
       sist(6) = X0 (initial state) =
    0.
    0.
       sist(7) = Time domain =
 c

[A,B,C,D]=abcd(sist);//Obtiene las Matrices del modelo de estado
s=poly(0,'s');
G1=C*inv(s*eye(2,2)-A)*B+D // Evalúa la fórmula (4.18)
G1  =
      1 + s
    ---------
              2
    6 + 5s + s
```

Obsérvese que la F.T. obtenida evaluando la fórmula (4.18) es corresponde perfectamente a la función de transferencia original. Si se obtiene mediante el comando `ss2tf` el resultado es el mismo:

```
G2=ss2tf(sist)
G2  =
      1 + s
    ---------
              2
    6 + 5s + s
```

Ejemplo de obtención de un modelo en variables de estado.

Consideremos el circuito RLC serie de la figura 4.3a, para el cual ya se obtuvo la función de

transferencia (4.8). Se desea ahora obtener un modelo en variables de estado.

Como ya se mencionó, hay una infinidad de modelos posibles, dependiendo de las variables de estado elegidas. Por esta razón, si no se especifica otra cosa ***es tradicional elegir como variables de estado del circuito: voltajes en capacitores y corrientes en inductores***, es decir, en este caso las variables de estado serán:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} v_c(t) \\ i(t) \end{bmatrix} \tag{4.19}$$

Aplicando la Ley de Voltajes de Kirchoff a la unica malla del circuito, se obtiene

$$v_L(t) + v_R(t) + v_C(t) = V_i(t) \tag{4.20}$$

es decir,

$$L\frac{di(t)}{dt} + Ri(t) + v_C(t) = V_i(t) \tag{4.21}$$

Además, se sabe que

$$v_C(t) = \frac{1}{C}\int_0^t i(t)dt \tag{4.22}$$

por lo tanto,

$$\frac{dv_C(t)}{dt} = \frac{1}{C}i(t) \tag{4.23}$$

despejando de (4.21)

$$\frac{di(t)}{dt} = -\frac{1}{L}v_C(t) - \frac{R}{L}i(t) + \frac{1}{L}V_i(t) \tag{4.24}$$

Expresando (4.23) y (4.24) en términos de las variables de estado definidas por (4.19) se obtiene

$$\dot{x}_1 = \frac{1}{C}x_2 \tag{4.25}$$

$$\dot{x}_2 = -\frac{1}{L}x_1 - \frac{R}{L}x_2 + \frac{1}{L}V_i(t) \tag{4.26}$$

Las dos anteriores son las ecuaciones de estado del circuito, sin embargo, es costumbre expresarlas en forma matricial como sigue

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V_i(t) \tag{4.27}$$

y la ecuación de salida es

$$y = x_1 \tag{4.28}$$

Es decir, las matrices A, b, c y d correspondientes a (4.12), (4.13) son las siguientes

$$A = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{R}{L} \end{bmatrix},\ b = \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix},\ c = \begin{bmatrix} 1 & 0 \end{bmatrix},\ d = 0 \tag{4.29}$$

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 4.4 se muestra en forma de bloque un sistema MIMO con p entradas y q salidas.

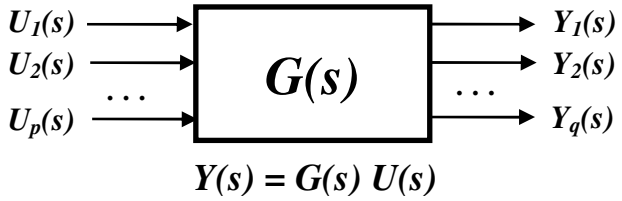

Figura 4.4.- Diagrama de Bloques de un sistema con p entradas y q salidas

Para el sistema MIMO de la figura 4.4 sigue siendo válida la relación

$$Y(s) = G(s)U(s) \tag{4.30}$$

sin embargo, (4.30) es ahora una expresión matricial donde:

$$Y(s) = \begin{bmatrix} Y_1(s) \\ Y_2(s) \\ \dots \\ Y_q(s) \end{bmatrix}, \quad U(s) = \begin{bmatrix} U_1(s) \\ U_2(s) \\ \dots \\ U_p(s) \end{bmatrix}, \quad G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \dots & G_{1p}(s) \\ G_{21}(s) & G_{22}(s) & \dots & G_{2p}(s) \\ \vdots & \vdots & \ddots & \vdots \\ G_{q1}(s) & G_{q2}(s) & \dots & G_{qp}(s) \end{bmatrix} \tag{4.31}$$

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

$$G_{ij}(s) = \left. \frac{Y_i(s)}{U_j(s)} \right|_{U_k(s)=0} \quad para\ k \neq j, \quad i = 1,2,...,q, \quad j = 1,2,...,p \tag{4.32}$$

Tanto el comando `syslin` como los comandos `ss2tf y tf2ss` trabajan con sistemas MIMO y por lo tanto manejan matrices de transferencia de manera directa.

Ejemplo: Para introducir en Scilab 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:

```
//define numeradores:
num11=1; num12=poly([0 1],'s','c');
num21=poly([2 1],'s','c'); num22=poly([0 1],'s','c');
//Define denominadores:
den11=poly([1 1 1],'s','c'); den12=poly([1 1],'s','c');
den21=poly([1 1],'s','c'); den22=poly([1 1 1],'s','c');
```

```
//Define Matriz de transferencia
G=syslin('c',[num11,num12;num21,num22],[den11,den12;den21,den22])
```

O también:

```
//define numeradores:
num11=1; num12=poly([0 1],'s','c');
num21=poly([2 1],'s','c'); num22=poly([0 1],'s','c');
//Define denominadores:
den11=poly([1 1 1],'s','c'); den12=poly([1 1],'s','c');
den21=poly([1 1],'s','c'); den22=poly([1 1 1],'s','c');
//Define componenentes de la matriz de transferencia
G11= syslin('c',num11,den11);  G12= syslin('c',num12,den12);
G21= syslin('c',num21,den21);  G22=syslin('c',num22,den22);
//Define la Matriz a partir de sus componentes
G=[G11,G12;G21,G22]
```

O también, usando la variable de Laplace:

```
s=poly(0,'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 tranferencia G(s) por cualquiera de los tres métodos anteriores, Scilab la desplegará como sigue:

```
G  =
         1            s
     ---------      -----
                2
     1 + s + s      1 + s

     2 + s             s
     -----          ---------
                              2
     1 + s          1 + s + s
```

Ejercicio: Escribir una <u>función</u> de Scilab en un archivo-M que a partir de los coeficientes de los polinomios del numerador (num) y denominador (den) de una F. T. obtenga la siguiente información:

- La función de transferencia (G)
- Los polos (p) y los ceros (z).
- Las matrices A,B,C,D del modelo en espacio de estado.

El comando para ejecutar la función deberá ser: `[G,p,z,A,B,C,D]=fun_trans(num,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_trans` 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 Scilab, de la forma G=FuncionTrans(R,L,C).