INSTRUCTIVO DE MATLAB

CÁTEDRA: Teoría de Control

AÑO : 2011

ÍNDICE

INTRODUCCIÓN

El interés en el estudio de la herramienta MATLAB versa en las facilidades que esta aplicación nos brinda en la manipulación de polinomios, el cálculo de funciones transferencia en lazo cerrado, la obtención de reducciones de los diagramas de bloques y el cálculo de la respuesta de un sistema a diversas entradas, entre otras. De esta manera la herramienta resulta un medio amigable para la corrección de problemas, la contrastación con resultados obtenidos en la teoría y la comprensión del funcionamiento de sistemas complejos debido a la flexibilidad de MATLAB para modificar diversas variables en forma inmediata.

FUNCIONES IMPORTANTES EN MATLAB

OPERACIONES ARTIMÉTICAS

En MATLAB se pueden realizar las operaciones aritméticas siguientes:

+	Sumar
-	Restar
*	Multiplicar
/	Dividir
^	Elevar a una potencia

```
>>5 +3

8
```

DECLARACIÓN DE UN VECTOR

Para declarar un vector fila, es suficiente escribir sus coordenadas entre corchetes

```
>> p = [1 3 0 4] %vector de 4 coordenadas

p=

1 3 0 4
```

>> p = [1 3 0 4]; %si se coloca punto y coma al final de la declaración, MATLAB no mostrará el valor de la variable recientemente declarada.

DECLARACIÓN DE UN POLINOMIO

Con MATLAB se puede trabajar con polinomios de forma sencilla. Es suficiente tener en cuenta que un polinomio no es nada más que un vector, en que el orden de los coeficientes va de mayor a menor grado.

```
>> p = [1 3 0 4] %1*x^3+3*x^2+0*x+4
```

```
p=1 3 0 4
```

Además, MATLAB incluye funciones específicas para operar con polinomios. Por ejemplo,

FUNCIONES ROOTS, POLY, CONV Y POLYVAL

La función roots permite obtener las raíces de un polinomio.

```
>> p = [1 3 0 4];
>> r = roots(p); %declaramos la variable r y almacenamos en ella las raíces del polinomio p
```

La función poly permite reconstruir un polinomio a partir de sus raíces.

Siguiendo con el ejemplo anterior,

```
>> p = poly(r); %reconstruimos p a partir de las raíces r
```

La función conv permite multiplicar dos polinomios,

Supóngase que se desea expandir el polinomio

$$n(s) = (3s^2 + 2s + 1)(s + 4)$$

```
>> p = [3 2 1]; q=[1 4];  %multiplica p y q
>> n = conv(p,q);  %declaramos n y le asignamos la multiplicación de p y q
```

La función polyval sirve para evaluar lo que vale un polinomio en un punto,

```
>> value = polyval (n,-5) %declaramos una variable value a la cual le asignamos el valor de n(s) en s=-5
```

DECLARACIÓN DE FUNCIONES TRANSFERENCIA

Para la introducción de funciones de transferencia polinómicas se utiliza la función “sys=tf(num,den)” del modo que a continuación se indica:

Introducir en Matlab la función de transferencia polinómica siguiente:

$$G(s) = G_1(s)+G_2(s) = \frac{s^2+12s+15}{s^3+3s^2+7s+5}$$

donde G1(s) y G2(s) son

$$G_1(s) = \frac{10}{s^2 + 2s + 5}$$

$$G_2(s) = \frac{1}{s + 1}$$

%Introducimos la función de transferencia polinómica G1(s)

```
>> num1=[10];den=[1 2 5];
```

```
>> sys1=tf(num1,den1)  %asigmanos a sys1 la función transferencia G1(s)
```

%Introducimos la función de transferencia polinómica G2(s)

```
>> num2=[1];den2=[1 1];
```

```
>> sys2=tf(num2,den2)  % asignamos  a sys1 la función transferencia G2(s)
```

%Sumamos ambas funciones transferencia para obtener G(s)

```
>> sys=sys1+sys2  % asignamos  a sys la suma de G1(s) y G2(s)
```

FUNCIONES POLE Y ZERO

La función pole permite localizar los polos de un sistema lineal

```
>> p=pole(sys)  %asignamos a p los polos de sys
```

La función zero permite localizar los ceros de un sistema lineal

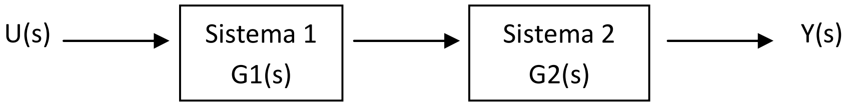
```
>> z=zero(sys)  %asignamos a z los ceros de sys
```

RESOLUCIÓN DE CONEXIONES SERIE, PARALELO Y REALIMENTACIÓN

FUNCIONES SERIES

La función series permite obtener un sistema equivalente a dos conectados en serie.

Conexión serie:

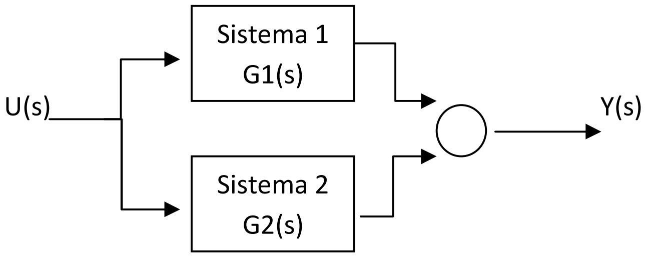


>> [sys]=series(sys1,sys2)

FUNCIONES PARALLEL

La function parallel permite obtener un sistema equivalente a dos conectados en paralelo.

Conexión paralelo:

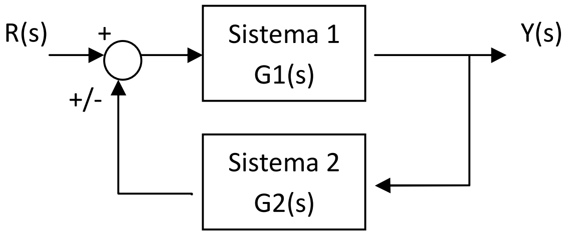


>> [sys]=parallel(sys1,sys2)

FUNCIONES FEEDBACK

La función feedback permite obtener un sistema equivalente cuando un sistema se encuentra realimentado por otro.

Conexión con realimentación:



>> [sys]=feedback(sys1,sys2,sign)

Obs: sign es realimentación negativa por defecto

<u>Ejemplo:</u>

Sea el proceso G(s) y el controlador H(s). Calcular la función transferencia de lazo cerrado. Considerar realimentación negativa.

$$G(s) = \frac{1}{500\, s^2}$$

$$H(s) = \frac{s+1}{s+2}$$

>> numg=[1]; deng=[500 0 0]; sys1=tf(numg,deng);

>> numh=[1 1]; denh=[1 2]; sys2=tf(numh,denh);

>> sys=feedback(sys1,sys2);

>> sys %devuelve function transferencia a lazo cerrado

RESPUESTA TRANSITORIA A UNA ENTRADA ESCALÓN E IMPULSO

Matlab permite la visualización en forma gráfica de la respuesta de un sistema ante distintos tipos de entradas. En este instructivo nos centraremos en las entradas escalón e impulso.

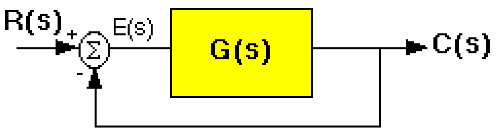
FUNCION STEP: ENTRADA TIPO ESCALÓN

La función step, permite obtener la respuesta transitoria de un sistema ante una entrada escalón de amplitud 1.

Ejemplo:

Dado el siguiente sistema, hallar la respuesta transitoria considerando una entrada escalón de amplitud 1.

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



% Introducir una función de transferencia polinómica

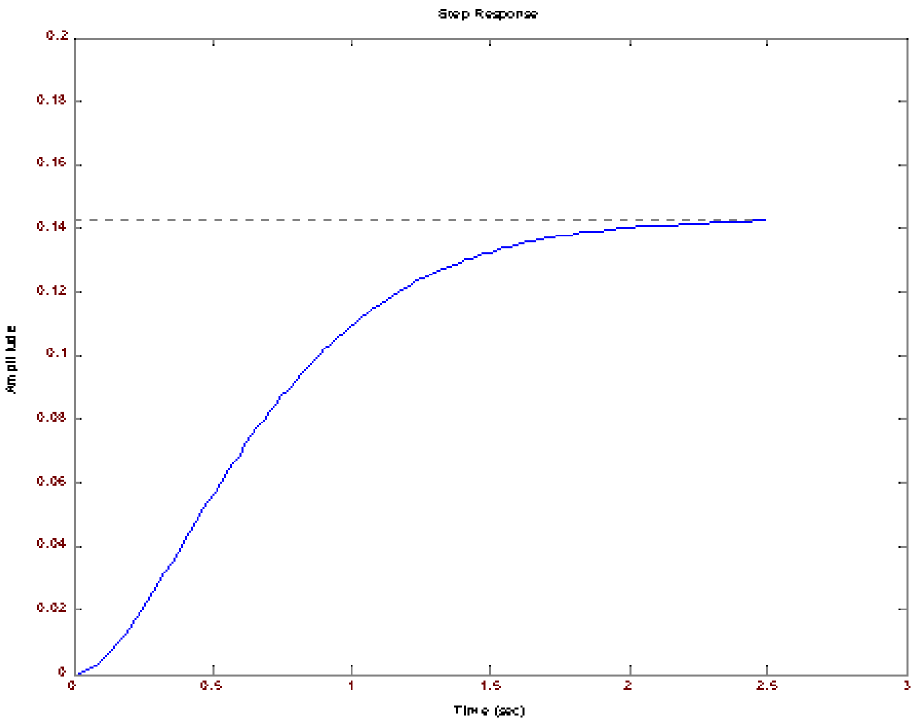
»num = 1;
»den = conv([1 2],[1 3]);

%Hallamos la función transferencia a lazo cerrado
» [clnum,clden] = cloop(num,den);

%Luego la respuesta transitoria del sistema resulta:

»step(clnum,clden)

Se obtendrá el gráfico de la respuesta transitoria



FUNCION IMPULSE: ENTRADA TIPO IMPULSO

La función impulse, permite obtener la respuesta transitoria de un sistema ante una entrada impulso.

Para obtener la respuesta de un sistema ante una entrada impulso se debe ejecutar la siguiente función.

% Obtener respuesta a una entrada impulso

» impulse(sys)

CONVERSIÓN DE MODELOS

MATLAB permite que distintos modelos se puedan convertir de la forma cociente de polinomios a la forma factorizada (ceros-polos-ganancia).

FUNCION RESIDUE

La función residue convierte la función de transferencia polinómica

$$H(s) = \frac{b_0 s^n + b_1 s^{n-1} + \ \dots \ + b_{n-1} s \ + b_n}{a_0 s^m + a_1 s^{m-1} + \ \dots \ + a_{m-1} s \ + a_m}$$

en la función transferencia de fracciones parciales:

$$H(s) = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \ \dots . + \frac{r_n}{s - p_n} + k(s)$$

Ejemplo:

Considérese la siguiente función de transferencia:

$$\frac{B(s)}{A(s)} = \frac{2s^3 + 5s^2 + 3s + 6}{s^3 + 6s^2 + 11s + 6}$$

%Introducimos el polinomio del numerador y del denominador de la función transferencia

»num=[2 5 3 6];

»den=[1 6 11 6];

%Luego ejecutamos la instrucción residue con num y den como parámetros

» [r,p,k]=residue(num,den)

Proporciona el resultado siguiente:

```
[r,p,k] = residue (num,den)

r=

        -6.0000

        -4.0000

        3.0000

p=

        -3.0000

        -2.0000

        -1.0000

k=

        2
```

(Observe que los residuos se devuelven en el vector columna r, las posiciones de los polos en el vector columna p y el término directo en el vector fila k). Ésta es la representación en MATLAB del siguiente desarrollo en fracciones simples de B(s)/A(s):

$$\frac{B(s)}{A(s)} = \frac{2s^3 + 5s^2 + 3s + 6}{s^3 + 6s^2 + 11s + 6} = \frac{-6}{s+3} + \frac{-4}{s+2} + \frac{3}{s+1} + 2$$

La función residue también se puede utilizar para obtener los polinomios (numerador y denominador) a partir de su desarrollo en fracciones simples.

» [num,den]=residue(r,p,k)

donde r,p y k están como se obtienen en el resultado de MATLAB anterior, convierte el desarrollo en fracciones simples en la razón de polinomios B(s)/A(s) del modo siguiente:

```
[num,den]=residue(r,p,k);

printsys(num,den,'s')

num/den=
```

$$\frac{2s\text{^}3+5s\text{^}2+3s+6}{s\text{^}3+6s\text{^}2+11s+6}$$

La función

»printsys(num,den,'s') % imprime num/den en términos del cociente de los polinomios en s.

EJERCICIOS PROPUESTOS

1)

Obtenga el desarrollo B(s)/A(s) siguiente en fracciones simples utilizando MATLAB.

$$\frac{B(s)}{A(s)}=\frac{s^2+2s+3}{(s+1)^3}=\frac{s^2+2s+3}{s^3+3s^2+3s+1}$$

Obtenga la función original a partir de la que se obtiene en fracciones simples.

2)

Resuelva la siguiente ecuación diferencial:

$$\ddot{x}+2\dot{x}+10x=t^2, \qquad x(0)=0, \qquad \dot{x}(0)=0$$

3)

Obtenga una solución analítica y una solución computacional de la respuesta escalón unitario de un sistema de realimentación unitaria cuya función de transferencia en lazo abierto es.

$$G(s)=\frac{5(s+20)}{s(s+4.59)(s^2+3.41s+16.35)}$$

Hallar el error en estado estable para una entrada escalón unitario y una entrada rampa unitaria.

4)

Para el siguiente sistema con realimentación unitaria determínese los errores en estado estacionario para una entrada escalón y una entrada rampa cuando

$$G(s)=\frac{10}{(s^2+14s+50)}$$

ANEXO: Método de la división larga para obtener la transformada Z inversa

TRANSFORMADA Z INVERSA:

Tiene por objetivo obtener la secuencia de señales en tiempo discreto representada por la transformada Z, para lo cual se debe obtener la transformada Z inversa. Se representa:

$$Z^{-1}\{F(z)\} = f[k]$$

Se puede obtener de varias maneras:

1. Por tabla
2. Mediante Fracciones Simples
3. Expansión como una serie de potencias mediante división larga

En este instructivo nos centraremos en el tercer método:

EXPANSIÓN COMO UNA SERIE DE POTENCIAS MEDIANTE DIVISIÓN LARGA

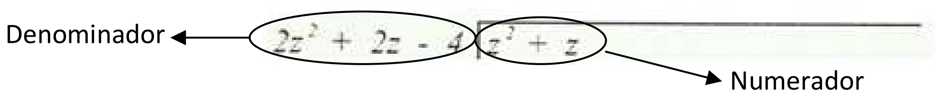
Se verá el método a través de un ejemplo clásico:

Ejemplo:

Obtener mediante división larga, la transformada z inversa de la función F(z).

$$F(Z) = \frac{(z+1)z}{2(z-1)(z+2)} = \frac{z^2+z}{2z^2+2z-4}$$

Colocamos el numerador y el denominador de la siguiente forma:



Ambos polinomios se deben ordenar de mayor a menor grado y se deben completar.

Luego se debe dividir el primer término del numerador con el primer término del denominador, es decir $\frac{z^2}{2\,z^2}$=0.5. Este resulta ser el primer término de la serie de potencias buscada.

$$\begin{array}{l|l} & 0.5 \\ \cline{2-2} 2z^2+2z-4 & z^2+z \end{array}$$

Luego se multiplica el denominador por este primer término del cociente y se resta al numerador.

$$\begin{array}{l|l} & 0.5 \\ \cline{2-2} 2z^2+2z-4 & z^2+z \\ & z^2+z-2 \end{array}$$

$$\begin{array}{l|l} & 0.5 \\ \cline{2-2} 2z^2+2z-4 & z^2+z \\ & \underline{z^2+z-2} \\ & \qquad\quad 2 \end{array}$$

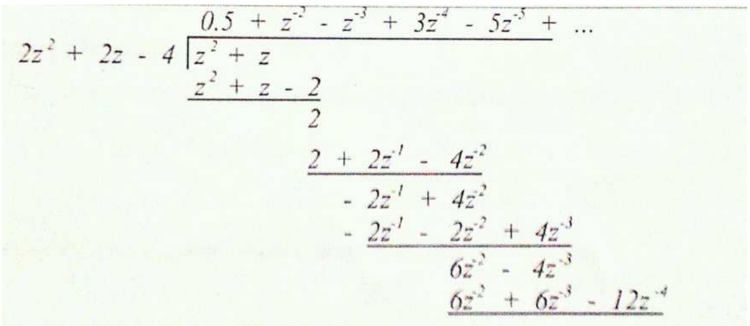
Luego se divide el resto, en este caso 2 con el primer término del denominador, es decir $\frac{2}{2\,z^2} = z^{-2}$, para luego multiplicar el denominador por este término, quedará:

$$\begin{array}{l|l} & 0.5+z^{-2} \\ \cline{2-2} 2z^2+2z-4 & z^2+z \\ & \underline{z^2+z-2} \\ & \qquad\quad 2 \\ & \qquad\quad 2+2z^{-1}-4z^{-2} \end{array}$$

Se realiza nuevamente la resta,

$$\begin{array}{l|l} & 0.5+z^{-2} \\ \cline{2-2} 2z^2+2z-4 & z^2+z \\ & \underline{z^2+z-2} \\ & \qquad\quad 2 \\ & \qquad\quad \underline{2+2z^{-1}-4z^{-2}} \\ & \qquad\quad -2z^{-1}+4z^{-2} \end{array}$$

Se repite el procedimiento hasta obtener el siguiente resultado:



Como vemos se ha logrado expresar como serie de potencias de la siguiente manera:

$$F(Z)=\frac{(z+1)z}{2(z-1)(z+2)}=\frac{z^2+z}{2z^2+2z-4}=\frac{1}{2}+\frac{1}{z^2}-\frac{1}{z^3}+\frac{3}{z^4}-\frac{5}{z^5}+\cdots$$

de manera tal que

f(k)=0.5,1,-1, 3,-5,…

que representa la secuencia en tiempo discreto.

<u>Observación:</u> La división se debe realizar hasta que la serie de potencias obtenida siga una secuencia determinada que no requiera continuar con la división para obtener el resto de los términos.