

UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL BUENOS AIRES

DEPARTAMENTO SISTEMAS

Catedra: TEORIA DE CONTROL

Trabajo Práctico de Laboratorio:

Uso de Matlab para Sistemas de Control

Características del entorno Matlab

MATLAB es un paquete de software de ingeniería que se utiliza ampliamente y que proporciona un entorno potente y amigable para cálculo y simulación en ingeniería. El entorno de programación ofrece operaciones matemáticas básicas más una serie de procedimientos operacionales (conocidos como funciones) que son llamadas por una única orden en MATLAB. Están disponibles muchas diferentes "toolboxes" las cuales extienden aún más las funciones básicas del Matlab a diferentes áreas; en estos tutoriales haremos uso extensivo de la toolbox de control de sistemas (Control Systems Toolbox. Otra herramienta de modelado que se ofrece con el paquete de MATLAB es el SIMULINK.

Ventanas de Matlab

Una vez ingresado al programa deberá aparecer el indicador >>, que nos dice que Matlab está esperando que introduzcamos un comando.

Matlab utiliza tres ventanas de exhibición: la ventana de comandos sirve para introducir comandos y datos e imprimir resultados; la ventana de gráficos sirve para exhibir curvas y gráficas, y la ventana de edición sirve para crear y modificar archivos M, que son archivos que contienen un programa o guión de comandos Matlab.

Cuando entramos en Matlab, la ventana activa es la de comandos, es posible escribir programas o guiones en la misma, pero estos se borrarán al salir de Matlab, en cambio si los mismos se generan en archivos M se pueden recuperar simplemente tecleando el nombre del archivo sin su extensión. Luego tecleando:

run 'nombre del archivo-m' se ejecutarán los comandos guardados en dicho archivo.

Comandos principales utilizados en control

abs	Valor absoluto
axis	Corrige la escala del gráfico actual, vea también plot , figure
bode	Dibuja el diagrama de Bode, vea también logspace, margin , nyquist1
conv	Convolución (útil para multiplicar polinomios), vea también deconv
deconv	Deconvolución y división de polinomios, vea también conv
dimpulse	Respuesta al impulso de sistemas lineales de tiempo discreto, vea también dstep
dstep	Respuesta al escalón de sistemas lineales de tiempo discreto, vea también stairs
feedback	Conexión de dos sistemas por realimentación.
figura	Crea una nueva figura o redefine la figura actual , vea también subplot, axis
for	Lazo For-Next
function	Para archivos-m del tipo función
grid	Dibuja la grilla en el gráfico actual
gtext	Agrega texto al gráfico actual, vea también text
help	Ayuda
hold	Mantiene el gráfico actual, vea también figure
if	Ejecuta código condicionalmente
imag	Devuelve la parte imaginaria de un número complejo, vea también real
impulse	Respuesta al impulso de sistemas lineales de tiempo continuo, vea también step , lsim , dlsim
inv	Inversa de una matriz
legend	Leyenda en un gráfico
length	Largo de un vector, vea también size
log	logaritmo natural, también log10: logaritmo común
lsim	Simula un sistema lineal, vea también step , impulse , dlsim .
norm	Norma de un vector
plot	Dibuja un gráfico, vea también figure, axis, subplot.
poly	Devuelve el polinomio característico

polyadd	Suma dos polinomios
polyval	Valor numérico de un Polinomio
print	Imprime el gráfico actual (a impresora o a archivo postscript)
real	Devuelve la parte real de un número complejo, vea también imag
rlocus	Grafica el lugar de raíces
sqrt	Raíz cuadrada
roots	Raíces de un polinomio
step	Dibuja la respuesta al escalón , vea también impulse , lsim , dlsim .
subplot	Divide la ventana Gráfico en secciones, vea también plot , figure
text	Agrega texto al gráfico actual, vea también title , xlabel , ylabel , gtext
tf	Crea una función de transferencia o convierte a función de transferencia,
tf2zp	representación Función de Transferencia a Polo-cero , vea también zp2tf
title	Agrega un título al gráfico actual
zp2tf	Polo-cero a representación función de transferencia , vea también tf2zp

Vectores

Comenzamos creando algo simple, como un vector. Entre cada elemento del vector (separado por un espacio) y entre corchetes, y asígnelo a una variable. Por ejemplo, para crear el vector **a**, entre en la ventana de comandos de Matlab (puede copiar y pegar):

- `a = [1 2 3 4 5 6 9 8 7]`

Matlab devolverá: (ejecute el comando y pegue el resultado)

Si ahora se quiere crear un vector con elementos desde 0 a 20 a incrementos de 2 (esto se usa mucho en la creación de vectores tiempo):

- `t = 0:2:20`

t=

- Suponga que hay que sumar 2 a cada elemento en el vector 'a'. La ecuación que lo logra se ve:

- $b = a + 2$

b=(ejecute el comando y pegue el resultado)

- Ahora suponga que quiere sumar dos vectores. Si los mismos tienen igual longitud, es sencillo. Simplemente súmelos como se muestra abajo

- $c = a + b$

c=(ejecute el comando y pegue el resultado)

- La resta de vectores de la misma longitud trabaja exactamente de la misma manera.

Matrices

Para ingresar una matriz numérica puede hacerse de dos maneras

Matriz=[1 2 3

4 5 6

7 8 9]

matriz=[1,2,3;4,5,6;7,8,9]

Operaciones con matrices: sume multiplique y reste las matrices anteriores

Copie y pegue el resultado

Funciones

Para facilitar el tratamiento de operaciones y fórmulas, Matlab incluye muchas funciones estándar. Cada función es un bloque de código que lleva a cabo una tarea específica. Matlab contiene todas las funciones estándares como **sin**, **cos**, **log**, **exp**, **sqrt**, así como tantas otras. También incorpora las constantes comúnmente usadas como π , e i o j para la raíz cuadrada de -1.

- $\sin(\pi/4)$

(ejecute el comando y pegue el resultado)

plot

Es muy sencillo crear figuras en Matlab, Suponga que necesita plotear una onda senoidal como función del tiempo . Primero defina un vector tiempo (el punto y coma luego de cada sentencia obliga a Matlab a no mostrar los resultados) y luego calcule el valor del seno en cada momento.

```
t = 0:0.25:7;
```

```
y = sin(t); plot(t,y);
```

(ejecute el comando y pegue el resultado)

subplot

Permite dividir la ventana de graficación en subventanas y situar las gráficas en cada una de ellas

Subplot(m,n,p)

Subdivide la ventana en m filas y n columnas y p es la ubicación

Actividad 1

Graficar la función $y = e^x$ con x natural y menor o igual a 7 en dos gráficos uno con escala lineal y otro con eje y en escala semilogarítmica.

Actividad 2

Guardar lo anterior como archivo exponencial.m llevar el dominio hasta 10

Y ejecutar desde el command utilizando el comando **run**

Valores Complejos

Matlab reconoce el operador "j" e "i" ejemplo

$C=30+20*j$

$\text{modulo}=\text{abs}(c)$

$\text{argumento}=\text{angle}(c)$

$\text{grados}=\text{argumento}*180/\pi$

imag

real

Pegar resultados

Polinomios

En Matlab, un polinomio se representa por un vector. Para crear un polinomio, simplemente entre cada coeficiente del mismo en un vector en orden descendiente (no omita el 0). Por ejemplo, quisiera entrar el siguiente polinomio:

$$s^4 + 3s^3 - 15s^2 - 2s + 9$$

Para hacerlo en Matlab, nada más entre el vector

- $x = [1 \ 3 \ -15 \ -2 \ 9]$

Matlab puede interpretar un vector de longitud $n+1$ como un polinomio de grado n . Por lo tanto, si el polinomio no tiene algunos coeficientes, es necesario que entre los ceros apropiados en el vector. Por ejemplo,

$$s^4 + 1$$

se representaría en Matlab como:

$x=$

Puede hallar el valor numérico de un polinomio con la función **polyval**. Por ejemplo, para hallar el valor en $s=2$ en el polinomio de arriba,

- $z = \text{polyval}([1 \ 0 \ 0 \ 0 \ 1], 2)$

Actividad 3

Hallar el valor del polinomio $3s^5 + 2s^3 - 162s^2 + 12$ para $s = -3$

RAÍCES

Hallar la raíz del polinomio

$$s^4 + 3s^3 - 15s^2 - 2s + 9$$

Para hallar las raíces hay que ingresar el siguiente comando;

- roots

ans =

Actividad 4

Halle las raíces del polinomio de la actividad 3

Se puede construir un polinomio conociendo sus raíces usando el comando

poly

Reconstruya el polinomio cuya raíces son -1, 2, 3

Función de transferencia

Para llevar a cabo una función de transferencia con MATLAB, debemos ingresar los polinomios numerador y denominador, en forma de vectores de la siguiente manera:

```
num=[a b];
```

```
den=[c d e];
```

Observemos que para definir el vector lo hacemos colocando entre corchetes los coeficientes de cada término, ordenados de mayor orden al menor. Para separar las columnas del vector lo hacemos con un espacio, o también podríamos utilizar coma.

Recuerde que el punto y coma final es para que el resultado de lo ejecutado por MATLAB no salga por pantalla.

Polos, ceros y ganancia

Si ahora ingresamos:

```
[z,p,k]=tf2zp(num,den)
```

Obtenemos:

z= cero de la función

p= polos de la función

k= ganancia de la función

Dado que toda función transferencia dada por un cociente de polinomios se puede escribir de la forma

$$G(s) = k \frac{\prod_{i=1}^m (s - z_i)}{\prod_{i=1}^n (s - p_i)} \quad \text{con } m \leq n,$$

Si queremos realizar el procedimiento inverso, necesitamos ingresar las variables k, z y p. Con la instrucción:

```
[num,den]=zp2tf(z,p,k);
```

Si queremos que MATLAB arme la función de transferencia como cociente de productos de los ceros y los polos, para ello utilizamos zpk, de la siguiente forma:

```
G=zpk(z,p,k);
```

Actividad 5

Ingresa la siguiente función de transferencia que tiene como numerador a $3s^2 + 5s + 20$ y denominador a $s^4 + s^2 + 4s + 12$ y hallar los ceros polos y amplificación, luego ingresar una función de transferencia que tiene polos en 1, 2, 3 y 4, ceros en 0 y 1 y ganancia 8 hallar los polinomios numerador y denominador, definir la función de transferencia G y usar el comando **printsys(num,den)** o **g=tf(num,den)** para visualizar la función transferencia.

Desarrollo en Fracciones simples

Cuando analizamos un sistema de control, por lo general disponemos de su función transferencia a lazo cerrado $G(s)$, donde

Con lo que podemos escribir la salida en función de la transferencia y la entrada:

$$\Theta_{o(s)} = \Theta_{i(s)} \cdot G(s)$$

Si deseáramos conocer la respuesta temporal $\Theta_o(t)$ del sistema cuando lo excitamos con una señal de entrada $\Theta_i(t)$, debemos calcular la transformada inversa de Laplace.

Como sabemos, es más sencillo de antitransformar cuando se trata de un cociente de polinomios, dado que si lo expresamos en fracciones simples podemos antitransformar directamente.-1

$$G(s) = \frac{16s + 16}{(s + 2)(s + 4)} \quad \text{y que} \quad \Theta_i(s) = \frac{1}{s}$$

Como las raíces del denominador son reales y distintas, el método de desarrollo en fracciones simples nos permite escribir a $\Theta_{i(s)} \cdot G(s)$ de la siguiente manera:

$$\begin{aligned} c(t) &= \mathcal{L}^{-1} \left\{ \frac{A}{s} + \frac{B}{s+2} + \frac{C}{s+4} \right\} \\ &= \mathcal{L}^{-1} \left\{ \frac{A}{s} \right\} + \mathcal{L}^{-1} \left\{ \frac{B}{s+2} \right\} + \mathcal{L}^{-1} \left\{ \frac{C}{s+4} \right\} \\ &= A + Be^{-2t} + Ce^{-4t} \end{aligned}$$

En general, estos cálculos pueden tornarse muy complicados de realizar 'a mano'. Veamos como se simplifican utilizando la función MATLAB `residue`. Ingrese los polinomios numerador y denominador de la misma forma como lo venimos haciendo hasta ahora. Ingrese ahora la sentencia:

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

Actividad 6

utilizando el comando residue hallar la $f(t)$ sabiendo que $\text{num}=[0 \ 1 \ 2]$ y $\text{den}=[4 \ 5 \ 6 \ 8]$.

Actividad 7

Sabiendo que una $f(t)$ posee res en -6, con polo en -3; -4 con polo en -2; 3 con polo en 1, +2 con polo en el origen. Hallar la función como cociente de polinomios, utilizando el comando

$[\text{num}, \text{den}]=\text{residue}(\text{res}, p, k)$; en donde k es el valor del res en polo 0

Sistemas de Primer Orden

La ecuación diferencial que describe la evolución de la temperatura en un termómetro sumergido en un líquido que se encuentra a la Temp. de un termómetro Θ_1

$$\tau \frac{d\Theta_2}{dt} + \Theta_2 = \Theta_1$$

Aplicando la transformada de Laplace, la transmitancia será

$$G = \frac{\Theta_2}{\Theta_1} = \frac{1}{\tau s + 1}$$

Actividad 7

Usando el editor de archivos m crear una aplicación que grafique la respuesta a un escalón unitario para distintos valores de τ . como ser $\text{tau}=1:0.4:4$

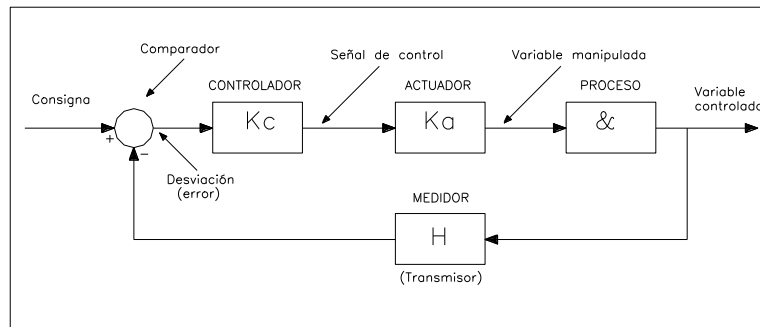
Utilizando los comandos $y=\text{step}(\text{num}, \text{den})$ y `for- end`, `hold on` y `grid`.

Que conclusión obtiene del gráfico?

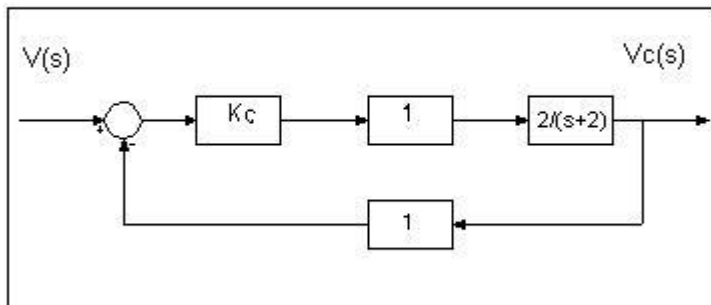
Sistema de Control a lazo cerrado con controlador p

En un proceso de primer orden y su expresión será de la forma

$$G = \frac{v(c)}{v(t)} = \frac{1}{\tau s + 1}$$



hacemos $K_a=1$ y $H=1$ y utilizando $2/(s+2)$ como planta nos queda:



Actividad 8

Utilizando Matlab realizar un programa que grafique la evolución de la función de primer orden siguiente con realimentación unitaria y controlador proporcional (como la explicada anteriormente) de amplitud K_c , la cual varía 1 a 10 si se le aplica una entrada escalón unitario utilice el comando **step(num,den)**. Utilice los comandos **for**, **end**, para ir introduciendo los valores K_c , utilice el comando **grid** para marcar una grilla en el gráfico, y **hold on** para mantener las curvas dibujadas

$$v_c(t) = \frac{K_c}{0.5s + K_c + 1} =$$

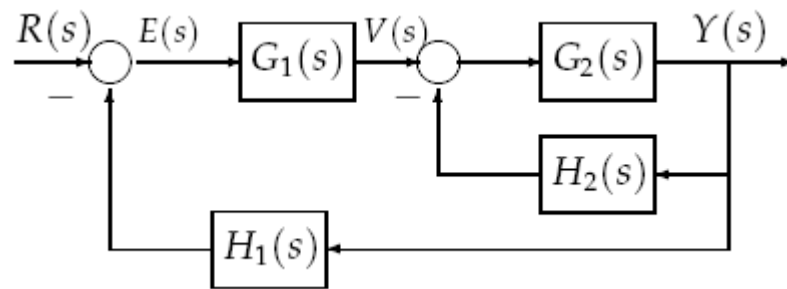
Saque una conclusión acerca del error en estado estable.

Función transferencia a lazo cerrado- Transferencia Global-

Supongamos que disponemos del sistema de la Figura siguiente donde pretendemos hallar la función de transferencia a lazo cerrado

$$\mathbf{G(s) = Y(s)/R(s) .}$$

Si aplicamos reducción de bloques, o resolviendo el diagrama de bloques



En MATLAB la función transferencia a lazo cerrado se puede calcular de dos formas:

- _ Utilizando SIMULINK (lo veremos próximamente).
- _ Utilizando las funciones de MATLAB series, parallel, feedback y cloop.

Para calcular la función transferencia a lazo cerrado $G(s)$ sigamos los siguientes pasos:

1. Definimos los numeradores y denominadores de las funciones transferencia de cada bloque de la siguiente forma:

```
numg1=; deng1=;
numg2=; deng2=;
numh2=; denh2=;
```

2. Calculamos la función transferencia de $V(s)$ a $Y(s)$:

```
[numvc,denvc]=feedback(numg2,deng2,numh2,denh2,-1);
```

3. Ahora calculamos la función transferencia de $E(s)$ a $Y(s)$ con:

```
[numec,denec]=series(numg1,deng1,numvc,denvc);
```

4. Calculamos el lazo cerrado:

```
[num, den]=cloop(numec,denec,- 1);
```

5. luego utilizamos printsys para verificar la solución.

Actividad 9

Para el sistema anterior y dados los siguientes valores hallar la función de transferencia global utilizando los comandos Matlab y el procedimiento anteriormente descrito.

$$G1(s) = 0.4;$$

$$G2(s) = 100/s(s+2) ;$$

$$H2(s) = s/s+20$$

$$H1(s) = 1;$$

Verificar que la misma sea

$$G(s) = 40s + 800 / s^3 + 22s^2 + 180s + 800$$

Respuesta de un sistema de segundo orden

Actividad 10

Dado el sistema de segundo orden siguiente

$$G(s) = Y(s)/R(s) = 4 / s^2 + 0.8s + 4$$

Utilizando los comandos step e impulse graficamos las respuestas. Hallamos la respuesta a la rampa unitaria y analizamos los resultados.

Actividad 11

Sabiendo que el factor que acompaña a la s determina el amortiguamiento del sistema y, tomando la función de transferencia anterior, realizar un programa en MATLAB que dibuje una familia de curvas para valores de este factor comprendidos entre 0:0.25:1.2 con respuesta al escalón y al impulso. Establecer una conclusión.