

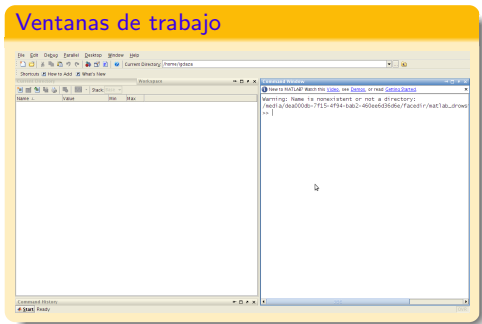
Práctica 1: Introducción a Matlab. Toolbox de Control

Automática Básica

Departamento de Automática, Universidad de Alcalá.

Escritorio de Matlab

Ventanas de trabajo



- Command Windows
 - Ejecuta comandos o funciones
 - Visualiza resultados
- Current Directory
 - El directorio de trabajo actual
- Workspace
 - Lista las variables, vectores y matrices definidas
- Command History
 - Historial de las funciones y comandos ejecutados.

Ventana de comandos

- Órdenes en línea.

```
>> 2+4  
ans = 6
```

- Ayuda en línea

```
>> help  
Ayuda sobre los distintos comandos.
```

```
>> help plot  
Ayuda sobre un comando específico.
```

```
>> lookfor palabra  
Muestra todas las órdenes relacionadas con "palabra"
```

- Formatos de salida. Precisión numérica

```
>> help format  
format short: Coma fija con 5 dígitos  
format long: Coma fija con 15 dígitos  
format short e: Coma flotante con 5 dígitos  
format long e: Coma flotante con 15 dígitos  
format hex: Hexadecimal  
format rat: Se aproxima al entero más pequeño
```

Variables

- Todas las variables son consideradas como matrices.
- Deben comenzar con una letra.
- Distingue mayúsculas de minúsculas.
- No se usarán caracteres de puntuación.
- Algunas de las variables predefinidas en Matlab:
 - **ans** Variable de resultado por defecto
 - **pi** Número pi
 - **i, j** $i = j = \sqrt{-1}$
 - **realmax** El mayor número real positivo utilizable
 - **realmin** El menor número real positivo utilizable
 - **inf** Infinito
 - **NaN** No número, por ejemplo 0/0

Archivos *.m (Scripts y funciones)

- Son archivos de texto plano ASCII
- Los archivos de programa en Matlab deben tener extensión “*.m”
- Deben comenzar con una letra
- Distingue mayúsculas de minúsculas
- No se deben usar caracteres de puntuación
- Son secuencias de instrucciones de Matlab y estructuras para el control del flujo de las mismas
- Se pueden realizar funciones dentro de él. Aparece la palabra **function** en la primera línea del fichero.
- Dentro de un fichero .m se puede llamar a otro .m o a sí mismo recursivamente.
- Las líneas de comentario comienzan por %.

Operadores

Aritméticos

- + Suma
- - Resta
- * Multiplicación
- / División
- ^ Potencia
- ' Transpuesta

Relación

- < Menor que
- > Mayor que
- <= Menor o igual que
- >= Mayor o igual que
- == Igual
- ~= Distinto, no igual

Lógicos

- & AND
- | OR
- ~ NOT

Funciones científicas:

```
>> help elfun (librería de funciones elementales)
```

```
...
```

```
Exponential.
```

```
exp - Exponential.
```

```
log - Natural logarithm.
```

```
log10 - Common logarithm.
```

```
sqrt - Square root
```

```
...
```

Vectores y matrices

- Vector fila

```
>> fila = [4 5 3]
fila = 4 5 3
>> v = 1:2:10
v = 1 3 5 7 9
>> v = 5:10
v = 5 6 7 8 9 10
```

- Elemento de un vector

```
>> v(3)
ans = 7 ;Elemento 3 del vector v
```

- Vector columna

```
>> col = [1; 3; 6]
col =
     1
     3
     6
```

- Matriz 2 x 3

```
>> matriz=[1 2 3; 4 5 6]
matriz =
     1     2     3
     4     5     6
```

- Elemento de una matriz

```
>> matriz(2,2)
ans = 5 ;Elemento (2,2)
de matriz
```

Operaciones con matrices (I)

Escalares:

- + Suma
- - Resta
- * Multiplicación
- / División

Ejemplo:

```
>> p = [1 2 3]
```

```
p =
```

```
1     2     3
```

```
>> o = 3*p
```

```
o =
```

```
3     6     9
```

Elemento a elemento. Deben tener el mismo número de elementos:

- + Suma
- - Resta
- .* Multiplicación
- ./ División
- .^ Potencia

Ejemplo

```
>> q = o+p
```

```
q =
```

```
4     8    12
```

```
>> q = o.*p
```

```
q =
```

```
3    12    27
```


Operaciones con matrices (II)

Transpuesta

- Para matrices reales: ' o bien .'
- Para matrices complejas
 - Transpuesta compleja conjugada: '
 - Transpuesta compleja no conjugada: .'

Ejemplos:

```
>> p = [1 2 3]
p = 1 2 3
>> q = [4 8 12]
q = 4 8 12
>> r=p+q*i
r = 1.00 + 4.00i 2.00 + 8.00i 3.00 + 12.00i
```

```
>> s=r'
s = 1.00 - 4.00i
      2.00 - 8.00i
      3.00 - 12.00i
>> t=r.'
t = 1.00 + 4.00i
      2.00 + 8.00i
      3.00 + 12.00i
```

Polinomios

- Su tratamiento es similar a vectores fila:
`>> p=[1 24 0 3 12]` ;ejemplo para el polinomio: $x^4+24x^3+3x+12$
- `roots`; Calcula las raíces de un polinomio
`>> r=roots(p)`
- `poly`; Construye el polinomio a partir de las raíces.
`>>p= poly(r)`
- `polyval`; Determina el valor de un polinomio para un valor dado.
`>>y= polyval(p,x)`
- `conv`; Multiplica dos polinomios
`>> a=[1 3 4];`
`>> b=[3 4 5];`
`>> conv(a,b)`
`ans = 3 13 29 31 20`
- `deconv`; Divide dos polinomios
`>> deconv(a,b)`

Control de flujo (I)

Bucles **for**: Ejecuta una serie de órdenes un número determinado de veces:

```
for variable=vector  
sentencias  
end
```

Ejemplo:

```
for k=1:5  
    x(k)=k^2 ;  
end  
>> x  
x= 1 4 9 16 25
```

Condiciones **if..else**: Evalúa una condición para ejecutar unas órdenes u otras:

```
if condicion  
sentencias  
end
```

Ejemplo:

```
altura = 80;  
if altura == 80  
    peso = 40;  
else  
    peso = 50;  
end
```

Función PLOT (I)

- **plot(x,y)**
 - Dibuja un gráfico en 2D.
 - x e y son dos vectores con el mismo número de elementos.
 - Une las distintas parejas de puntos (xn, yn).
- **plot(x,y,'s')**
 - 's' es una cadena de 1, 2 ó 3 caracteres, que permite definir el color y/o el tipo de línea para el trazado.
 - **plot(a,b,'s1',c,d,'s2',...)**
 - Dibuja dos o más gráficos superpuestos.

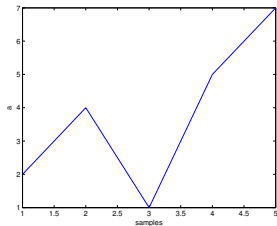
Tipo de trazo	Estilo de línea	Color de línea
. puntos	- continua	y amarillo
o círculos	: de puntos	m violeta
x aspas	-. punto y guión	c celeste
+ cruces	- discontinua	r rojo
...

Función PLOT (II)

- **plot(Y)**

Si los elementos de Y son reales y no se pone el vector x, dibuja el gráfico en función del número de índice de cada elemento de y (1,2,3,...).

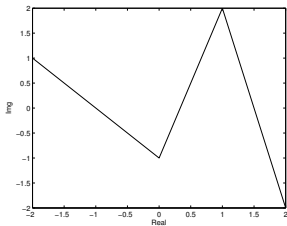
```
>> a=[2 4 1 5 7];  
>> plot (a, 'b')
```



- **plot(Y)**

Si los elementos de Y son complejos, dibuja el gráfico en el plano complejo (parte real en abscisas, parte imaginaria en ordenadas).

```
>> b=[-2+j -j 1+2j 2-2j];  
>> plot (b, 'k')
```



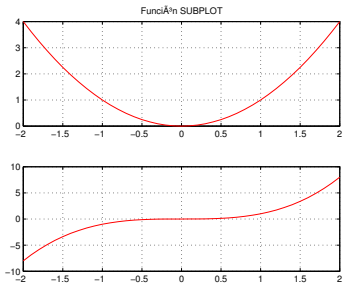
Funciones relacionadas

- **axis([xmin xmax ymin ymax])**
Controla la escala de los ejes.
- **axis('auto')**
Devuelve a los ejes la escala automática (por defecto)
- **title('Cadena_de_texto')**
Dibuja el título del gráfico
- **xlabel('Cadena_de_texto'). ylabel('Cadena_de_texto')**
Pone una etiqueta en los ejes X y/o Y.
- **grid on|off**
Dibuja una regilla.
- **hold on|off**
Activa|desactiva la 'retención' del gráfico anterior
- **loglog(x,y)**
Establece ejes logarítmicos
- **semilogx(x,y)**
Eje x logarítmico.
- **semilogy(x,y)**
Eje y logarítmico.

Función SUBPLOT

- **subplot(m,n,p)**
 - Crea una matriz de gráficos de 'm' filas y 'n' columnas en una misma ventana gráfica.
 - 'p' es el número de orden de cada gráfico.

```
x=-2:0.1:2;  
subplot(2,1,1);  
plot(x,x.^2,'r')  
grid on  
title('Función SUBPLOT')  
subplot(2,1,2);  
plot(x,x.^3,'r')  
grid on
```



Método para modular código

- Son ficheros .M que comienzan por la palabra *function*.
- El nombre del archivo deben comenzar con una letra.
- Debe ir en minúsculas y no usar caracteres de puntuación.
- Se le pueden pasar argumentos cuando se va a ejecutar.
- Las variables definidas dentro de la función son locales, desaparecen cuando se termina de ejecutar.
- Se pueden compilar y almacenar en librerías (Toolbox).
- Ejemplo para la función 'media()' en el archivo 'media.m' `function y=media(x)`

```
%Calcula la media de un vector
```

```
[m,n]=size(x);
```

```
if m==1
```

```
y=sum(x)/n;
```

```
else
```

```
error('debes introducir un vector, no una matriz');
```

```
end
```

```
>> z=[1:99];
```

```
>> a=media(z)
```

```
a = 50
```


Características y descripción de funciones (I)

- Características

- Para indicar a MATLAB la función de transferencia de un sistema se utiliza un polinomio en s para el numerador $N(s)$ y otro para el denominador $D(s)$. En la ayuda de Matlab, aparecen como 'NUM' y 'DEN'.
- NUM y DEN serán por tanto vectores que representan los polinomios numerador y denominador.

- Funciones

- **pzmap(num,den)**

- Dibuja el mapa de polos y ceros de un sistema cuya función de transferencia sea $G(s)=\text{num}(s)/\text{den}(s)$.

- **printsys(num,den)**

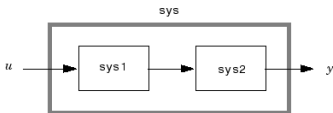
- Representa la función de transferencia del sistema $G(s)=\text{num}(s)/\text{den}(s)$ en forma de fracción.
- Únicamente es una mejora visual, más cercana a nuestra representación habitual.

Descripción de funciones (II)

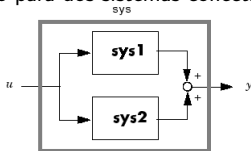
- **[z,p,k] = tf2zp(num,den)**
 - Dada la función de transferencia $G(s)=\text{num}(s)/\text{den}(s)$, calcula los polos 'p', los ceros 'z' y la ganancia 'k' de la misma.
 - 'z' y 'p' son dos vectores columna que guardan las raíces del polinomio 'num' y 'den' respectivamente.
 - 'k' es un escalar que guarda la ganancia del sistema.
- **[num,den] = zp2tf(z,p,k)**
 - Realiza la operación inversa a la anterior. Dada una función de transferencia en forma de polos, ceros y ganancia (z,p,k), calcula la misma en forma de polinomios en s, (num,den).
 - 'z' y 'p' deben ser vectores columna y 'p' un escalar.
 - 'num' y 'den' son vectores fila.

Descripción de funciones (III)

- **[num,den] = series(num1, den1, num2, den2)**
 - Dada las funciones de transferencia de dos sistemas $G1(s)=num1(s)/den1(s)$ y $G2(s)=num2(s)/den2(s)$ conectados en serie, sirve para calcular la función de transferencia del sistema equivalente (num,den).
 - num1, den1, num2, den2, num y den son vectores fila que representan polinomios en s.

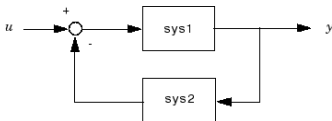


- **[num,den] = parallel(num1, den1, num2, den2)**
 - Igual que el anterior pero para dos sistemas conectados en paralelo.

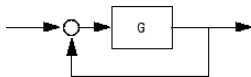


Descripción de funciones (IV)

- **[num,den] = feedback(num1, den1, num2, den2, signo)**
 - Calcula la función de transferencia de un sistema en lazo cerrado con realimentación no unitaria $G_2(s)=\text{num2}(s)/\text{den2}(s)$.
 - El signo indica si la realimentación es positiva o negativa.



- **[num,den] = feedback(num1, 1, signo)**
 - Calcula la función de transferencia de un sistema en lazo cerrado con realimentación unitaria.
 - El signo puede ser 1 ó -1, indicando si la realimentación es positiva o negativa.



Práctica 1: Introducción a Matlab. Toolbox de Control

Automática Básica

Departamento de Automática, Universidad de Alcalá.