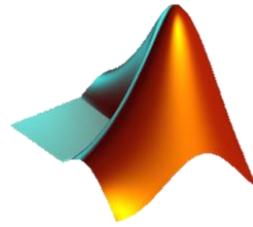


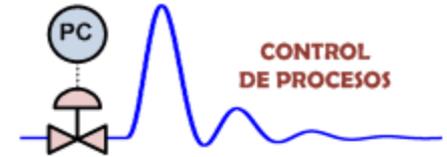
---

**CONTROL DE PROCESOS**

# **Introducción a MATLAB**



# MATLAB

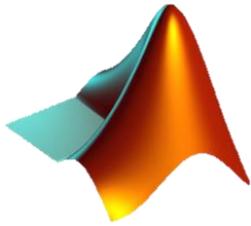


---

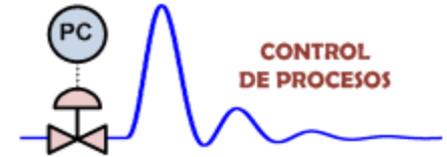
**MATLAB** surge en 1984 como un paquete de cálculo numérico matricial escrito en FORTRAN. Actualmente MATLAB está escrito en C, JAVA y ensamblador. Se comercializa por parte de la empresa **MathWorks** (<http://es.mathworks.com>). Es un programa multiplataforma, capaz de ejecutarse sobre arquitecturas Windows, MAC y GNU Linux.

Con el paso de los años, MATLAB ha ido creciendo para abordar distintos problemas de ingeniería, economía, etc. Estas nuevas rutinas se distribuyen en los llamados **Matlab Toolboxes**. MATLAB, en la actualidad, cuenta además con entornos gráficos de programación como el caso de **Simulink**.

---



# ENTORNO DE MATLAB



Ingreso a Simulink

Paleta de comandos

Directorio actual

Ventana de comandos

The screenshot shows the MATLAB 7.9.0 (R2009b) environment. The Command Window contains the following code and output:

```
>> A = rand(4,4)
A =
    0.8147    0.6324    0.9575    0.9572
    0.9058    0.0975    0.9649    0.4854
    0.1270    0.2785    0.1576    0.8003
    0.9134    0.5469    0.9706    0.1419

>> B = A*A'
B =
    2.8966    2.1881    1.1965    2.1551
    2.1881    1.9966    0.6827    1.8861
    1.1965    0.6827    0.7590    0.5348
    2.1551    1.8861    0.5348    2.0955

>> eig(B)
ans =
    0.0009
    0.1441
    0.7379
    6.8648
```

The Workspace window shows the following table:

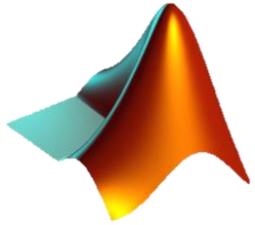
Name	Value	Min
A	<4x4 double>	0.0975
B	<4x4 double>	0.5348
ans	[9.3639e-04;0.1441;0.7379;6.8648]	0.0009

The Command History window shows the following commands:

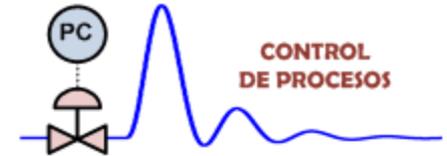
```
margin(g1)
g1 = 2/(s+1)^3;
margin(g1)
G = tf([1 2],[1 3 5])
bode
bode(G)
g1 = 2/(s+1)^3;
[MG,MF,wc,w1] = margin(g1)
ltiview
help ord2
14/02/17 10:25 --%
A = rand(4,4)
B = A*A'
eig(B)
```

Espacio de trabajo

Ventana del historial



# VARIABLES Y MATRICES

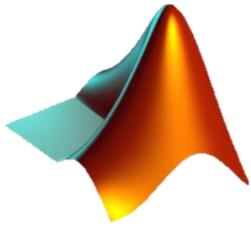


## Definición (=)

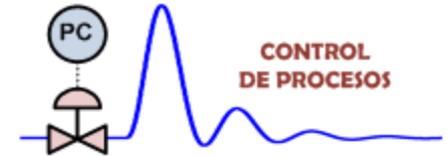
- >> A=1
- >> B=3; → No muestra la operación (;)
- >> C=A+B
- >> t=0:0.1:10; → Vector desde 0 hasta 10

## Matrices:

- >> A=[1 2 3 4 5 6] → Matriz 1x5
- >> B=[1,2,3,4,5,6] → Matriz 1x5
- >> C=[1 2 3;4 5 6] → Matriz 2x3
- >> D=C' → (Matriz Transpuesta) Matriz 3x2



# POLINOMIOS



Los polinomios se introducen como vectores fila que contienen los coeficientes de orden mayor a menor

>> p=[1 -5 8 3] **—————> equivale al polinomio  $s^3-5s^2+8s+3$**

Para calcular las raíces de un polinomio se define un vector con todos coeficientes del mismo y se aplica el comando roots:

```
>> p=[1 -5 8 3]
```

```
p =
```

```
1 -5 8 3
```

```
>> roots(p)
```

```
ans =
```

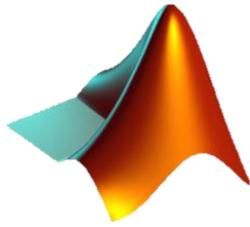
```
2.6554 + 1.6123i
```

```
2.6554 - 1.6123i
```

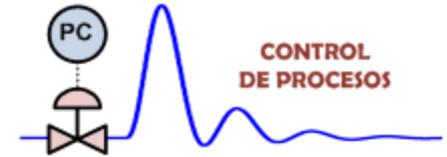
```
-0.3109
```

Puede hacerse lo mismo directamente con

```
>> roots([1 -5 8 3])
```



# POLINOMIOS



Se pueden multiplicar (**conv**) y dividir polinomios (**deconv**).  
Los resultados son vectores.

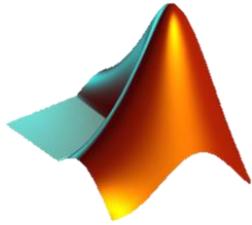
```
>> a = [1 2 3];  
>> b = [4 5 6];  
>> c = conv(a,b)  
c =  
    4    13    28    27    18  
>> [q,r] = deconv(c,a)  
q =  
    4    5    6  
r =  
    0    0    0    0    0
```

**$a(s) = s^2+2s+3$**

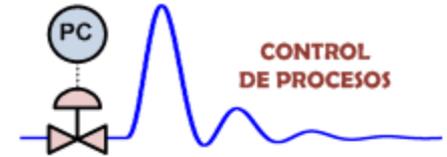
**$b(s) = 4s^2+5s+6$**

**Producto del polinomio  $a(s)$  por  $b(s)$  y el resultado en  $c$**

**División de  $c(s)$  en  $a(s)$  El resultado es el cociente  $q(s)$  y el resto  $r(s)$**



# OPERACIONES



Las operaciones aritméticas de Matlab se realizan de dos formas:

- ❑ **Como matriz.** Las operaciones siguen la regla del álgebra lineal
- ❑ **Como arreglo.** Las operaciones se hacen elemento a elemento (elementwise).

OPERACIÓN	TIPO	
	MATRIZ	ARREGLO
Adición	+	+
Sustracción	-	-
Multiplicación	*	.*
División Derecha	/	./
Exponenciación	^	.^

```
>> A = [0 2 -3];
```

```
>> B = [-2 1 5];
```

```
>> A*B'
```

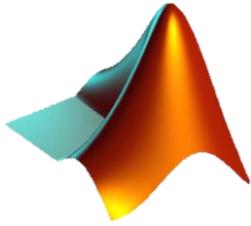
```
ans =  
-13
```

```
>> A.*B
```

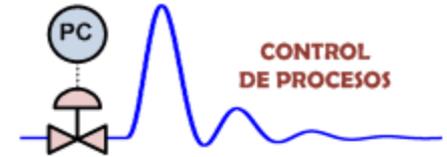
```
ans =  
0 2 -15
```

**Operación como matriz.**  
**Vector A por el transpuesto de B**

**Operación como arreglo.**  
**Multiplicación elemento a elemento del vector A por el B**



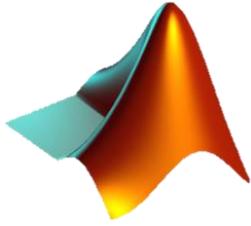
# FUNCIONES



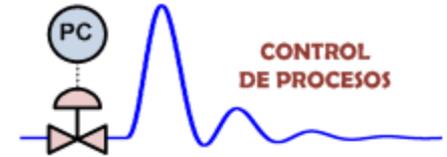
Matlab dispone de funciones para cálculos matemáticos donde el argumento puede ser un escalar, un vector o una matriz. Dispone de funciones trigonométricas, exponenciales, etc.

```
>> a = 2;  
>> A = [0 2 -3];  
>> exp(a) ← Exponencial de un escalar  
ans =  
    7.3891  
>> exp(A) ← Exponencial de un vector  
ans =  
    1.0000    7.3891    0.0498 ← Devuelve un vector
```

La lista de posibilidades la puede consultar con el comando ***help elmat.***



# OPERACIONES



Si se quiere encontrar la función  $y(t)$

$$y(t) = t e^{-t}$$

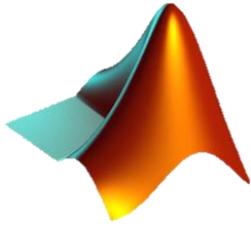
para valores de  $t$  entre 0 y 50 espaciados en 0.1 se puede seguir la secuencia:

```
>> t = 0:0.1:50;
```

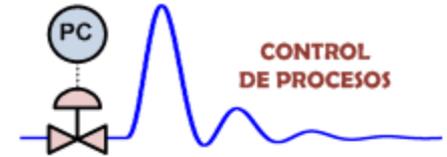
```
>> y = t.*exp(-t);
```

Define el vector **t** con elementos de 0 a 50:  
[0 0.1 0.2 ... 49.9 50]

Multiplicación elemento a elemento del vector **t** por el vector  $e^{-t}$



# GRÁFICAS 2D



La función **plot** genera una gráfica 2D en ejes cartesianos en una nueva ventana. Los argumentos de **plot** son: un vector de las abscisas y un vector correspondiente a la ordenada.

*Workspace*

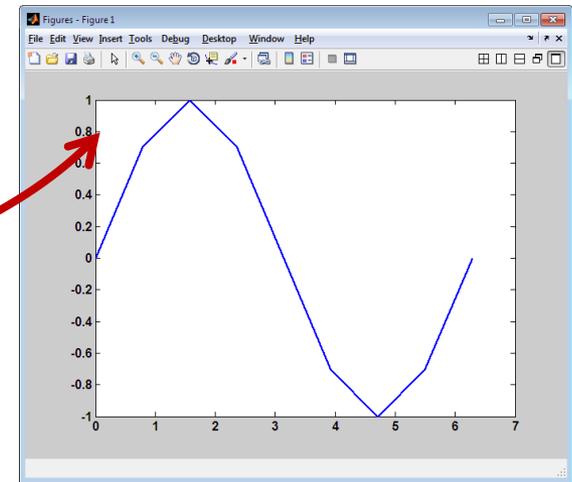
x	y
0	0
0.7854	0.7071
1.5708	1.0000
2.3562	0.7071
3.1416	0.0000
3.9270	-0.7071
4.7124	-1.0000
5.4978	-0.7071
6.2832	-0.0000

>> x = 0:pi/4:2\*pi;

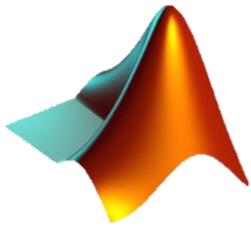
>> y = sin(x);

>>

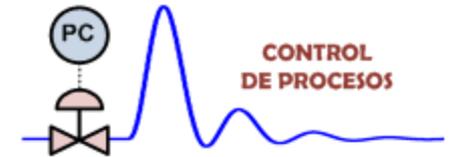
>> plot(x,y)



*Trabajando en la ventana Figure puede personalizar el gráfico añadiendo referencias, cuadriculando, etc.*



# GRÁFICAS 2D



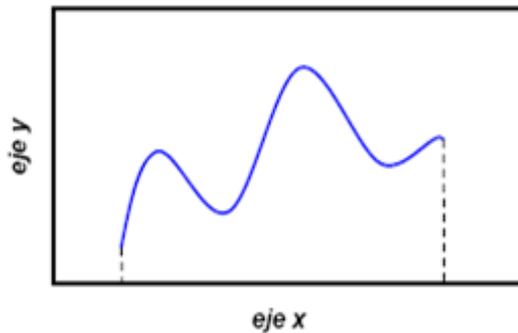
■ `plot( x , y )`

$$\begin{matrix} 1 \\ 2 \\ \vdots \\ n_1 \end{matrix} \begin{bmatrix} \cdot \\ \cdot \\ \vdots \\ \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \\ \vdots \\ \cdot \end{bmatrix}$$

Se pueden trazar más de una curva sobre los mismos ejes.

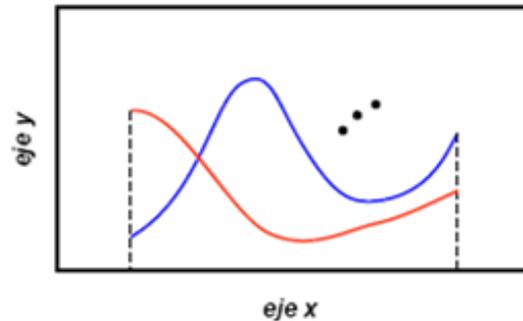
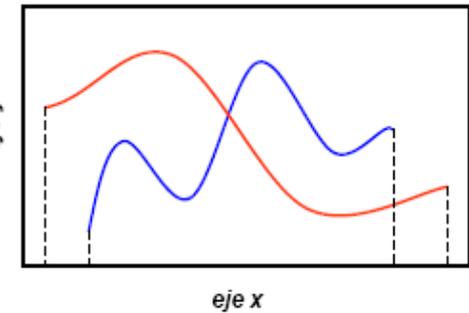
■ `plot( x1, y1 , x2, y2 , ... )`

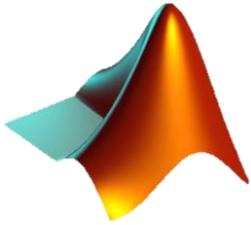
$$\begin{matrix} 1 \\ 2 \\ \vdots \\ n_1 \end{matrix} \begin{bmatrix} \cdot \\ \cdot \\ \vdots \\ \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \\ \vdots \\ \cdot \end{bmatrix} \begin{matrix} 1 \\ 2 \\ \vdots \\ n_2 \end{matrix} \begin{bmatrix} \cdot \\ \cdot \\ \vdots \\ \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \\ \vdots \\ \cdot \end{bmatrix}$$



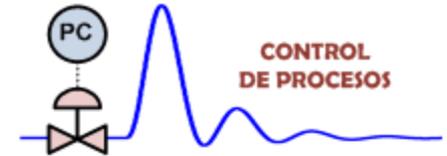
■ `plot( x , Y )`

$$\begin{matrix} 1 \\ 2 \\ \vdots \\ n_1 \end{matrix} \begin{bmatrix} \cdot \\ \cdot \\ \vdots \\ \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \vdots & \vdots \\ \cdot & \cdot & \dots \end{bmatrix}$$





# ESPACIO DE TRABAJO



**Command Window**

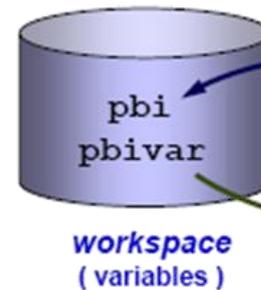
```
>> x = [35.56 34]
x =
    35.5600    34.0000
>> y = [-15-12] + x
y =
    8.5600    7.0000
fx >>
```

**Workspace**

Name	Value	Min	Max
x	[35.5600 34]	34	35.5600
y	[8.5600 7]	7	8.5600

El **Workspace** almacena las variables en arreglos similares a los de una hoja de cálculo.

Pueden transferirse datos del espacio de trabajo a una hoja de cálculo y viceversa.



Año	PBI (mil. S/ de 1994)
1950	21928.82203
1951	23987.07981
1952	25230.66146
...	...
2007	174329.1186
2008	191479.0896