



UNIVERSIDAD NACIONAL EXPERIMENTAL DEL TACHIRA  
DEPARTAMENTO DE INGENIERIA ELECTRONICA  
NUCLEO DE INSTRUMENTACION CONTROL Y SEÑALES  
LABORATORIO DE INSTRUMENTACION Y CONTROL

INTRODUCCION AL MATLAB. (PARTE I)

MATLAB es un ambiente de computación técnica de alto rendimiento, para cómputo numérico y visualización, que integra análisis numérico cómputo de matrices, procesamiento de señales y gráficas en un ambiente fácil de usar, donde los problemas y sus soluciones se expresan como se haría en matemáticas, sin necesidad de la programación tradicional.

MATLAB, cuyo nombre es una contracción de MATrix LABoratory, es un sistema interactivo cuyo elemento básico es una matriz que no requiere dimensionamiento. Esta característica le permite al usuario resolver muchos problemas numéricos en una fracción de tiempo.

MATLAB ha evolucionado a lo largo de los años con aportes de desarrolladores de muchos usuarios. En el ambiente educativo se ha venido usando desde el nivel de enseñanza media. En el ambiente universitario, MATLAB se ha convertido en una herramienta estándar para cursos introductorias en álgebra lineal, así como para cursos básicos y avanzados en diferentes áreas del conocimiento. En ambientes industriales, MATLAB se usa para investigar y para resolver problemas prácticos de ingeniería y problemas matemáticos.

MATLAB también ofrece una familia de aplicaciones que se han dado llamar en cajas de herramientas o “Toolboxes”. Un aspecto muy importante para la mayoría de los usuarios de MATLAB es que las cajas de herramientas son colecciones de funciones, códigos o macros de MATLAB, conocidos como M-files, muy fáciles de comprender y usar, que extienden el ambiente del programa con el fin de resolver clases particulares de problemas. Algunas de las áreas para las cuales existen Cajas de herramientas son: Procesamiento de Señales, Análisis y Diseño de Sistemas de Control, Simulación y Dinámica de Sistemas, Redes Neuronales, Ecuaciones Diferenciales Parciales, Estadística, Comunicaciones, Finanzas, Realidad Virtual, etc.

#### ASPECTOS GENERALES

- Los comandos son sensibles a mayúsculas y minúsculas: t, T, costo, Costo
- Toda variable debe empezar por letra.
- Para decimales se debe escribir con punto 2.45
- Se pueden separar comandos con coma(,) >>x1=12 ,x2=5
- Al colocar punto y coma al final(;) matlab ejecuta la instrucción, pero no la muestra en pantalla.

Se puede calcular una expresión directamente o usando variables o constantes

Ejemplo:

```
>>5*exp(-1.5)*sin(2*pi/5*1.5-37*pi/180)
```

La respuesta será:

```
ans=1.0549
```

Cuando no se asigna una variable al realizar alguna operación, esta quedara guardada en una por defecto llamada ans.

La misma expresión pero usando variables

```
>>A=5;t=1.5;T=5;w=2*pi/T;tita=37*pi/180;
>>y=A*exp(-t)*sin(w*t-tita)
y=1.0549
```

En el caso anterior, se asigno a la variable y el resultado de la operación.

Se puede utilizar puntos suspensivos (...) para continuar en la línea siguiente

```
>>y=A*exp(-t)*...
sin(w*t-tita)

y=1.0549
```

**who:** Permite ver las variables en memoria.

**whos:** Permite ver el formato de cada variable.

**help (comando):** Muestra ayuda del comando. Ejemplo.

```
>>help format
desplegará las diferentes opciones para fijar el formato de salida de datos
```

## NUMEROS COMPLEJOS

Forma rectangular:

Letras i,j como sufijo, solo con números

```
>>c1=3-4j
```

Letras i,j acompañadas con asterisco (\*) para expresiones de números complejos. A continuación se da un ejemplo de crear un numero complejo con parte real y parte imaginaria.

```
>> c2=6+j*sin(0.5)
real(): captura la parte real de un número complejo
imag(): Captura la parte imaginaria de un número complejo
>>c2re=real(c2),c2im=imag(c2)
abs(): Determina la magnitud de un numero complejo.
angle(): ángulo en radianes.
>>c2mag=abs(c2),c2fase=angle(c2)*180/pi
```

## FORMATOS PARA DESPLEGAR NUMEROS

Por defecto los números se presentan enteros o real con 4 decimales.

Notación científica para valores grandes o pequeños.

Resultados usando el comando format para el valor de pi

```
format short          3.1416
format long           3.14159265358979
format bank           3.14
format rat             355/113
format compat:       elimina líneas en blanco
format:               recupera el despliegue por defecto .
```

## ALGUNAS FUNCIONES MATEMATICAS

sin	- Seno.
sinh	- seno hiperbólico.
asin	- seno inverso.
asinh	- seno hiperbólico inverso.
cos	- Coseno.
cosh	- coseno hiperbólico.
acos	- coseno inverso.
acosh	- coseno hiperbólico inverso.
tan	- tangente.
tanh	- tangente hiperbólico.
atan	- tangente inverso.
atanh	- tangente hiperbolico inverso.
sec	- secante.
sech	- secante hiperbólico.
asec	- secante inverso.
asech	- secante hiperbólico inverso.
csc	- cosecante.
csch	- cosecante hiperbólico.
acsc	- cosecante inverso.
cot	- cotangente.
acot	- cotangente inverso.

### EXPONENCIALES.

exp	- exponencial.
log	- logaritmo natural.
log10	- logaritmo base 10.
log2	- logaritmo base 2.
pow2	- potencia base 2.
sqrt	- raíz cuadrada.

### COMPLEJOS.

abs	- valor absoluto.
angle	- ángulo.
complex	- construye un numero complejo.
conj	- complejo conjugado.
imag	- parte imaginaria.
real	- parte real.

### REDONDEOS.

fix	- Redondea a cero.
floor	- Redondea a menos infinito.
ceil	- Redondea a mas infinito.
round	- Redondea al entero mas cercano.
sign	- Muestra si es positivo o negativo el número.

## VECTORES Y MATRICES

Los vectores se escriben entre corchetes separando los elementos por espacios en blanco o por comas

```
>> w=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1]
```

Se puede multiplicar por una constante

```
>> x=w*pi
```

Para determinar el seno de cada elemento del vector

```
>> y=sin(x)
```

Para extraer un elemento de un vector, se coloca entre paréntesis la posición en el vector del elemento a extraer. Ejemplo, del vector anterior (y) se desea ver solo el valor ubicado en la posición 5 y guardarlo a una variable k.

```
>>k=y(5)
```

Se puede extraer una parte del vector para crear otro, colocando la posición inicial y luego la posición final, separadas por dos puntos .

```
>>m=y(3:7)
```

El anterior ejemplo, crea un vector (m) con los valores del vector (y) desde la posición 3 hasta la 7.

Para extraer elementos de un vector en forma selectiva

```
>>q=y([2 6 3 8 9])
```

Para crear matrices, se colocan los elementos entre corchetes colocando los valores por espacios en blanco y separando las filas con punto y coma (;)

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

extraer el elemento de la fila 3 columna 2 de una matriz

```
>> r=a(3,2)
```

Los dos puntos en las matrices, indican el rango de elementos. Ejemplo extraer de la fila 2, los elementos de las columnas 2 y 3

```
>>a(2,2:3)
```

Para crear una matriz mágica, se coloca entre paréntesis el número de filas y columnas .

```
>> A=magic(5)
```

Selección de todos los elementos de la segunda fila

```
>>A(2, :)
```

Selección de todos los elementos de la segunda y tercera fila

```
>>A(2:3, :)
```

Selección de toda la columna 3

```
>>A(:, 3)
```

Selección de todas las filas de la columna 1 a la 3

```
>>A(:, 1:3)
```

Construcción de vectores usando notación de dos puntos(:)

formato (valor inicial : incremento : valor final). Ejemplo

Ejemplo: crear un vector con valores desde 0 hasta 1 en incrementos de 0.1

```
>>x=(0:0.1:1)
```

La función **linspace**(n1,n2,n) distribuye uniformemente **n** puntos lineales partiendo de **n1** y terminando en **n2**

```
>>q=linspace(0,5,50)
```

Apóstrofe ('), cambia la orientación de los vectores y matrices determina la traspuesta

```
>>magic(4),magic(4)'
```

**size:** determina el tamaño de una matriz o vector .

**Length:** longitud de un arreglo .

Operaciones entre escalar y vector

```
>>t=0:1:10
```

```
>>t1=t-2
```

```
>>t2=2*t-1
```

```
>>y=[t t1 t2 ]
```

Lo anterior se llama construcción de un vector por concatenación, es decir, se puede construir una matriz o un vector, añadiendo partes de diferentes vectores

Operaciones entre matrices, elemento a elemento

```
>>a=[1 2 3 4;5 6 7 8;9 10 11 12]
```

```
>>b=[1 1 1 1;2 2 2 2;3 3 3 3]
```

suma de los elementos

```
>>a+b
```

multiplicación punto a punto

```
>>a.*b
```

**rand(n):** creación de un vector aleatorio con n valores

**rand(f,c):** matriz aleatoria con f filas y c columnas

```
>>rand(3)
```

```
>>rand(4,2)
```

Para cambiar los elementos de un arreglo, se indica la posición y luego el valor que se desee cambiar

```
>>b
```

```
>>b(3,2)=9
```

Otras funciones que se pueden realizar con matrices se pueden observar con help matlab\matfun

Sum( ): suma los elementos de un arreglo

```
>>b
```

```
>>sum(b(1,:))
```

Suma todos los elementos de la primera fila

```
>> sum(b(:,2))
```

suma todas las filas de la segunda columna

## MANIPULACION DE GRAFICOS

Un comando usado para el manejo de gráficos es el la instrucción PLOT.

PLOT(X,Y): dibuja la gráfica de los valores correspondientes del vector X con los del vector Y.

Se puede manipular el color marca o símbolo de cada una de las gráficas, usando en forma general PLOT(X,Y,S), donde S es una cadena de caracteres preestablecidos según la siguiente tabla :

b	azul	.	punto	-	línea sólida
g	verde	o	circulo	:	línea punteada
r	rojo	x	x	-.	Línea y punto
c	cian	+	+	--	línea a trazos
m	magenta	*	estrellas		
y	amarillo	s	cuadrado		
k	negro	d	diamante		
		v	triangulo (abajo)		
		^	triangulo(arriba)		
		<	triangulo(izquierda)		
		>	triangulo(derecha)		
		p	pentágono		

Ejemplo

Plot(t,y,'yd')

Dibuja un gráfico en color amarillo con diamantes

### REJILLAS, MARCO DE EJES Y ROTULOS

- 1) Para incluir una rejilla usar el comando grid después del gráfico.
- 2) grid off, remueve la rejilla, grid sin argumento alterna estados on-off.
- 3) box off elimina los marcos de los ejes.
- 4) box on recupera el marco; box sin argumento. Alterna estados on-off
- 5) title('titulo de la gráfica'); permite incluir un título centrado en la parte superior, se escribe entre apostrofes el texto
- 6) xlabel('texto eje x'), ylabel('texto eje y') permite generar los rótulos de los ejes horizontal y vertical

### CONFIGURACION DE LOS EJES

- 1) El comando axis permite controlar la apariencia de los ejes.
- 2) axis([xmin xmax ymin ymax])modifica los límites del gráfico actual

ejemplo: escriba los siguientes comandos y observe cada uno de los cambios que ocurren en la grafica

```
>>x=linspace(0,2*pi,100);
```

```
>>y=sin(x);
```

```
>>w=cos(x);
```

dibujar en un mismo gráfico el seno y coseno de x

```
>>plot(x,y,x,w)
```

```

>>grid      %coloca una rejilla
>>title('Curvas de sen(x) y cos(x) ') %coloca titulo superior
>>xlabel('eje x, tiempo') % coloca el título en eje X
>>ylabel('Amplitud de las ondas') %coloca titulo en eje Y
>>axis([0 2*pi -1.5 1.5]) %modifica los ejes de la gráfica

```

### MANIPULACIÓN DE GRAFICOS

- 1) Figure(n) permite abrir una nueva ventana de figuras, conservando la última.
- 2) hold on permite dibujar nuevos gráficos en el mismo juego de ejes
- 3) hold off libera la ventana de la figura actual para un nuevo gráfico.
- 4) Subplot(m,n,k) divide la ventana de la figura actual en una matriz mxn áreas y selecciona para realizar el gráfico en el valor de k

Ejemplo

```

>>subplot(2,2,1)
>>plot(x,y)
>>axis([0 2*pi -1 1])
>>title('sen(x)')
>>subplot(2,2,2)
>>plot(x,w)
>>title('cos(x)')
>>a=2*sin(x).*cos(x);
>>subplot(2,2,3)
>>plot(x,a)
>>title('2.sen(x).cos(x)')
>>grid
>>b=sin(x).^2./x;
>>subplot(2,2,4)
>>plot(x,b)
>>title(' sin^2(x)/x ')

```

**ACTIVIDADES:** Realice las actividades solicitadas por el profesor.