

## Tutoriales de Control con Matlab



---

# Tutorial: Lo Básico de Matlab

[Vectores](#)

[Funciones](#)

[Ploteo](#)

[Polinomios](#)

[Matrices](#)

[Impresión](#)

[Uso de archivos.M en Matlab](#)

[Obtención de Ayuda en Matlab](#)

---

En esta parte del Tutorial se usarán los siguientes Comandos Matlab :

**plot polyval roots conv deconv polyadd inv eig poly**

**Nota:** Commandos Matlab No-estándares se ven **verdes**.

---

Matlab es un programa interactivo útil en computación numerica y visualización de datos; los ingenieros en control lo usan extensivamente en análisis y diseño. 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). Matlab corre en los entornos Unix, Macintosh, y Windows ;hay una versión disponible para estudiantes que corre solo en PCs.

Para más información acerca del Matlab, contáctese con [Mathworks](#).

Lo mejor que puede hacer con estos tutoriales es verlos en una ventana mientras está corriendo Matlab en otra . Así estará habilitado para rehacer todos los ploteos y cálculos que están en los tutoriales mediante la operación de copiar y pegar el texto que ve acá directamente en la ventana del Matlab o en un archivo.m.

## Vectores

Comencemos 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:

- $a = 1\ 2\ 3\ 4\ 5\ 6\ 9\ 8\ 7$

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):

- $t = 0:2:20$   $t = 0\ 2\ 4\ 6\ 8\ 10\ 12\ 14\ 16\ 18\ 20$

Manipular vectores es tan fácil como crearlos. Primero suponga que hay que sumar 2 a cada elemento en el vector 'a'. La ecuación que lo logra se ve:

- $b = a + 2$   $b = 3\ 4\ 5\ 6\ 7\ 8\ 11\ 10\ 9$

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

- $c = a + b$   $c = 4\ 6\ 8\ 10\ 12\ 14\ 20\ 18\ 16$

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

## Funciones

Para hacernos la vida más fácil, 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  $\pi$ ,  $e$  o  $j$  para la raíz cuadrada de -1.

- $\sin(\pi/4)$   $ans = 0.7071$

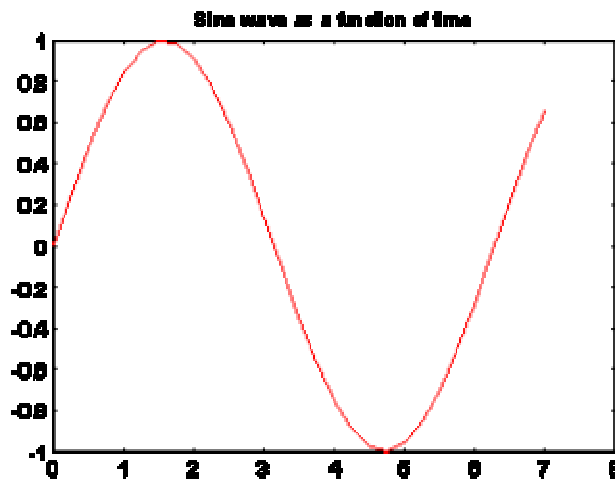
Para determinar la sintaxis y lo que hace la función sólo tipee `help [nombre_function]` en la ventana de comando de Matlab .

Matlab también le permite escribir sus propias funciones con el comando [function](#) ; siga este link para aprender cómo escribir sus propias funciones y aprecie un listado de las funciones que fueron creadas expresamente para este tutorial.

## Ploteo

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)`



La figura contiene aproximadamente un período de la onda senoidal. Este tipo de figuras es muy sencillo en Matlab, y el comando `plot` tiene muchas opciones agregadas. Se recomienda visitar la página [ploteo](#) para aprender más.

## 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

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

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:

- $y = [1 \ 0 \ 0 \ 0 \ 1]$

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

- $z = \text{polyval}([1 \ 0 \ 0 \ 0 \ 1], 2)$   $z = 17$

Puede también sacar las raíces de un polinomio. Esto es muy útil cuando se tiene un polinomio de alto grado como

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

Hallar las raíces es tan sencillo como entrar el siguiente comando;

- `roots([1 3 -15 -2 9]) ans = -5.5745 2.5836 -0.7951 0.7860`

Digamos que quiere multiplicar dos polinomios. El producto de dos polinomios se halla tomando la convolución de sus coeficientes. La función `conv` del Matlab lo hará .

- `x = [1 2]; y = [1 4 8]; z = conv(x,y) z = 1 6 16 16`

La división de dos polinomios es también sencilla. La función `deconv` devuelve tanto el resultado cuanto el resto. Dividamos `z` por `y`, y veamos si se obtiene `x`.

- `[xx, R] = deconv(z,y) xx = 1 2 R = 0 0 0 0`

Como puede ver , este es el mismo polinomio/vector `x` de arriba. Si `y` no es raíz de `z` , el vector `R` resto hubiese sido distinto de cero.

Si se quiere sumar dos vectores de igual orden, bastará la simple instrucción `z = x + y` (los vectores `x` e `y` deben ser de la misma longitud). En el caso general , debe usarse la función definida por el usuario, `polyadd` . Para usar `polyadd`, copie la función a un [archivo-m](#), y luego úsela como lo haría con cualquier función del Matlab. Asumiendo que ya tiene la función `polyadd` almacenada en un `archivo.m`, y quiere sumar dos polinomios no apareados, `x` e `y`, podrá hacerlo digitando la siguiente instrucción:

- `z = polyadd(x,y) x = 1 2 y = 1 4 8 z = 1 5 10`

## Matrices

Definir matrices en Matlab es lo mismo que entrar un vector, excepto que cada renglón de elementos debe separarse por un punto y coma(;). Ingrese :

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

`B =`

`1 2 3 4`

`5 6 7 8`

`9 10 11 12`

- `B = [ 1 2 3 4 5 6 7 8 9 10 11 12] B = 1 2 3 4 5 6 7 8 9 10 11 12`

En Matlab las matrices pueden manipularse de distintas maneras. Una de ellas: puede encontrar la traspuesta de una matriz de reales con la tecla apóstrofo:

- `C = B'`

`C =`

`1`

`2`

`...`

`12`

Debe notarse que si B hubiese sido complejo , Matlab hubiese devuelto la traspuesta conjugada. Para obtener siempre la traspuesta sólo, use .' (punto apostrofo, los dos comandos son iguales si la matriz no es compleja).

Ahora puede multiplicar dos matrices B y C . Recuerde que el orden tiene importancia en el producto de matrices.

- $D = B * C$  ,  $D = 650$  (la matriz de 1x1)  
 $D = C * B$  ,  $D =$  la matriz de 12x12

Otra opción en la manipulación de matrices es que se puede multiplicar los elementos correspondientes de dos matrices utilizando el operador .\* (punto-por, para hacer esto las matrices deben tener la misma dimensión).

- $E = [1 \ 2; 3 \ 4]$   $F = [2 \ 3; 4 \ 5]$   $G = E .* F$

$$E = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad F = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \quad G = \begin{bmatrix} 2 & 6 \\ 12 & 20 \end{bmatrix}$$

- Si se tiene una matriz cuadrada como la matriz E, puede efectuarse la potencia de E las veces que se quiera, lo que equivale a multiplicarla por sí misma  $n$  veces.

- $E^3$

$$ans = \begin{bmatrix} 37 & 54 \\ 81 & 118 \end{bmatrix}$$

Y si se quiere elevar al cubo cada elemento de la matriz basta con:

- $E.^3$

$$ans = \begin{bmatrix} 1 & 8 \\ 27 & 64 \end{bmatrix}$$

Para calcular la inversa de la matriz E:

- $X = \text{inv}(E)$

$$X = \begin{bmatrix} -2.0000 & 1.0000 \\ 1.5000 & -0.5000 \end{bmatrix}$$

o sus autovalores:

- $\text{eig}(E)$

$$ans = \begin{bmatrix} -0.3723 \\ 5.3723 \end{bmatrix}$$

Aún más, hay una función que encuentra los coeficientes del polinomio característico de una matriz. La función "poly" crea un vector con los coeficientes del polinomio característico en orden descendente.

- $p = \text{poly}(E) \quad p = 1.0000 \ -5.0000 \ -2.0000$

Recuerde que los autovalores de una matriz son los mismos que las raíces de su polinomio característico:

- $\text{roots}(p) \quad \text{ans} = 5.3723 \ -0.3723$

## Impresión

Imprimir en Matlab es sumamente sencillo. Sólo siga los pasos de abajo:

### Macintosh

- To print a plot or a archivo-m from a Macintosh, just click on the plot or archivo-m, select Print under the File menu, and hit return.

### Windows

- Para imprimir una figura o un archivo-m en una computadora bajo Windows, nada más seleccione Print en el menu File **de la ventana correspondiente al plot (la figura) o al archivo-m**, y haga enter.

### Unix

- To print a plot on a Unix workstation enter the command:  

```
print -P<printername>
```

 If you want to save the plot and print it later, enter the command:  

```
print plot.ps
```

 sometime later, you could print the plot using the command "lpr -P plot.ps" If you are using a HP workstation to print, you would instead use the command "lpr -d plot.ps"

To print a m-file, just print it the way you would any other file, using the command "lpr -P <name of m-file>.m" If you are using a HP workstation to print, you would instead use the command "lpr -d plot.ps<name of m-file>.m"

## Uso archivos-m en Matlab

Hay diferencias pequeñas que debe saber de cada plataforma.

### Macintosh

- Matlab trae un editor para archivos-m; elija "New M-file" en el menú File . También puede usarse el editor que quiera (lo único que debe asegurarse es que grabe los archivos en formato sólo texto y de cargarlo cada vez que empiece el Matlab ).

### Windows

- Correr Matlab bajo Windows es muy parecido a correrlo en una Macintosh. Sin embargo, debe saber que su archivo-m será salvado en el clipboard. Por lo que, debe asegurarse que está salvado como archivo.m

Unix

- You will need to run an editor separately from Matlab. The best strategy is to make a directory for all your m-files, then cd to that directory before running both Matlab and the editor. To start Matlab from your X- window, simply type: matlab.

Puede tipear los comandos directamente en el Matlab, o ponerlos en un archivo-m y entonces sólo correr el archivo-m. Si pone todos sus archivos.m en el mismo directorio en el que está instalado matlab, él lo encuentra siempre.

## Obtención de Ayuda en Matlab

Matlab tiene una muy buena ayuda en línea, simplemente hay que tipear :

- `help nombre_del_comando`

para mayor información sobre un determinado comando . Debe saber el nombre exacto del comando que busca; los links a [Lista de Comandos](#); en ésta y al final de cada página le brinda una lista de todos los comandos usados en este tutorial y en los ejemplos.

Aquí tiene algunas notas para terminar este tutorial.

Puede obtener el valor de cualquier variable en cualquier momento, tipeando su nombre. Por ejemplo B

```
B =  
 1 2 3  
 4 5 6  
 7 8 9
```

También puede escribir más de una instrucción en un renglón, sepárelos mediante una coma o un punto y coma.

Además seguramente se dió cuenta que de no asignar el resultado de un cálculo a una variable específica, Matlab lo pone en la variable "ans" .

---

## User feedback

We would like to hear about difficulties you had with los tutoriales, suggestions you have for improvement, errors that you found, or any other comments that you have. This feedback is anonymous; include your email address if you want a reply.

Hay que escribirlo en Inglés, ya que manda la pregunta o sugerencia al sitio original

---

## Tutoriales

Básicos | [Modelación](#) | [PID](#) | [Root Locus](#) | [Respuesta en Frecuencia](#) | [Espacio Estado](#) | [Digital](#) | [Ejemplos](#)

[Home](#)[Índice](#)[Comandos](#)

8/11/97 dmt