

# Procesamiento de Señales en MATLAB: Parte 1 - Introducción

---

## Señales y Sistemas

La progresión lógica en la utilización de MATLAB para resolver problemas de señales y sistemas es entender:

1. Qué tipo de números pueden ser representados
2. Cómo nombrar las variables y qué valores se les pueden asignar
3. Qué operadores matemáticos son construidos en MATLAB y cómo operan números y variables
4. Qué funciones matemáticas son intrínsecas en MATLAB
5. Cómo escribir un archivo script
6. Cómo escribir un archivo función para definir funciones matemáticas
7. Cómo mostrar los resultados de forma gráfica para obtener una rápida comprensión de relaciones entre variables

## Números, Variables y Matrices

MATLAB es basado en vector y matrices. Es decir, TODO en MATLAB es una matriz. Un vector de  $m$  elementos es una matriz  $1 \times m$  y un escalar es una matriz  $1 \times 1$ .

```
% Real numbers:  
a = 7.89  
b = 15.8e-11  
  
% Complex Numbers:  
c = 8+3*i  
d = 8+j*3  
% i y j estan preasignadas en MATLAB para ser iguales a raiz cuadrada de -1  
  
% constantes preasignadas en MATLAB:  
% i y j para raiz cuadrada de -1  
% pi  
% inf para infinito  
% NaN para not a number
```

## Operadores mas comunes

```
% = asignación  
% == igualdad  
% + adición  
% - substraccion y unary minus  
% * multiplicación de matrices  
% .* multiplicacion de arrays  
% ^ potencia de matrices
```

```
% .^ potencia de array
% / division
% ./ division de arrays
% <> operadores relacionales
% & AND logico
% ~ NOT logico
% | OR logico
```

## Prompt, ";" y whitespaces

```
% El caracter » indica que MATLAB espera pos instrucciones del usuario
»

% El punto y coma ; termina la instruccion
» a = 2 ;

% los espacios son ignorados por MATLAB

% ans expresa un resultado cuando el usuario no le asigno un nombre
% por ejemplo multiplicando a = 2 por b = 3
» a * b
ans = 6
```

## Vectores

```
% Asignación de vectores
b = [1 3 2] ; c = [1 , 5 , 5] ;
```

## Comandos

```
% para obtener las variables actualmente definidas y sus valores se utiliza
el comando who
» who
Your Variables are:
A  C  ans c
B  a  b

% para hacer indefinido el valor de una variable se utiliza el comando
clear
» clear A
» A
??? Undefined function or variable ==> A
```

## Multiplicacion

- Multiplicar una matriz por un escalar multiplica cada miembro del vector o matriz por el escalar.
- Sumar o restar un escalar a un vector o matriz, suma o resta el escalar a cada elemento del vector o matriz.
- Los vectores y matrices se suman (o restan) como normalmente se define en matemáticas. Es decir, los vectores o matrices deben tener la misma dimensión para realizar la operación.
- La multiplicación de vectores respeta las reglas del algebra. Por ejemplo

```

b = [1 2 3]
c = [3 2 1]
% multiplicar b*c retorna
» b * c
??? Error using ==> *
% Necesitamos utilizar la transpuesta de c para realizar la multiplicacion
de matrices
» b * c'
ans = 10
» b' * c
% esto retorna una matriz de 3x3.

% A veces es util multiplicar vectores o matrices de igual dimension
elemento por elemento. Este tipo de multiplicacion es llamada array
multiplication y se realiza mediante el operador .*

b. *c
ans =
    3    4    3

% Matrices

A = [ 3 5 1 ; 9 -1 2 ; -7 -4 3];
» A
A =
    3    5    1
    9   -1    2
   -7   -4    3

% En la potencia ^ se respeta las reglas del algebra
% para elevar elento por elemento se utiliza .^

```

## Division

- Se puede dividir escalares por escalar, vectores por escalar, matrices por escalar.
- Al utilizar A/B entre matrices es equivalente a multiplicar la matriz A por la matriz inversa de B.
- Se puede utilizar la division de array A. / B para dividir elemento por elemento
- Se puede dividir un escalar por una matriz utilizando el operador ./ por ejemplo 10./A nos devuelve una matriz donde cada elemento es el escalar dividido el elemento en la posicion correspondiente en la matriz

## Igualdad

El operador `==` compara dos numeros y devuelve el valor logico 1 si los numeros eran el mismo o 0 si no lo eran.

- Si dos vectores o dos matrices son comparadas deben tener la misma forma
- Se puede realizar con escalar con escalar, vector con vector, matriz con matriz y escalar con vector o matriz
- El operador `~=` devuelve 1 si no son iguales y 0 si son iguales

```
% operadores
% ~=
% >
% <
% >=
% <=
```

## Operadores Logicos

```
% & devuelve 1 si ambos operandos NO son cero y 0 cc
% & | devuelve 0 si ambos operandos son cero y 1 cc
% ~ devuelve 1 si el operando es 0, y 0 si el operando es distinto de cero
```

## Operador ":"

```
% Si ambos operandos son escalares, genera una secuencia separados por uno
» 3 : 8
ans = 3 4 5 6 7 8

% Utilizando la forma a:b:c donde todos son escalares genera una secuencia
de a a c separados por b
» -4:3:5
ans = -4 -1 2 5

% El incremento termina en el valor menor o igual que el valor especificado
c (cuando el inc es positivo)

% Otro uso: Vecotr Columna: la instruccion A(:) forma un vector columna con
todos los elementos en la matriz A

% Otro uso: Extraer submatriz: A(:,2) forma una matriz con la segunda
columna de A
» A(:,2)
ans =
    30
    23
   -11
```

```
»A(3, :)  
ans =  
    1 -11  2
```

## Scripts y Funciones

Existen dos tipos de programas en MATLAB, scripts y funciones. Ambos programas se almacenan con la extensión ".m".

Una función es un programa modular que acepta argumentos y devuelve resultados. El principal objetivo es que sea reutilizable en varias situaciones.

La primera línea en una función debe ser la palabra clave o *keyword* **function**

```
function y = nombreFuncion(a,b,c)
```

## Funciones y Comandos MATLAB

MATLAB tiene una lista de funciones y comandos intrínsecos para tareas comunes en matemáticas.

El comando *help* ofrece documentación de todos los comandos MATLAB.

```
» help abs  
  
ABS Absolute value  
(...)
```

## Comandos de control de flujo

### If

```
if a > b then  
    ...  
else  
    ...  
end
```

### For

```
for n = 1:15,  
    (...)  
end  
  
N = 16:4:0;
```

```
for n = N,
    (...)
end

% Tambien se pueden utilizar vectores
N = 1:10 ; x = [] ;
for n = []
    x = [x,n^2];
end
```

## While

```
while x > 32
    (...)
end
```

## Switch

```
switch expression
    case value1,
        ...
    case value2,
        ...
    ...
    case valueN
        ...
end
```

## Otros comandos

```
% Entrada de teclado
n = input('Message')M

% pausa hasta presionar una tecla y pausa por un tiempo n en segundos
pause;
pause(n);
```

## Strings

- La funcion char cnoiverte integers en su correspondiente caracter ASCII.
- **concatenación:** `strcat(s1,s2);`
- **comparación:** `strcmp(s1,s2);`
- **búsqueda:** `findstr(s1,'palabra');`
- **conversión de tipos:** `num2str(n)` y `str2num(s)`

## Archivos IN/OUT

```
% fopen
% apertura de archivos para lectura o escritura

% el argumento especifica el nombre y opcionalmente el path del file

% devuelve un integer positivo que identifica el archivo.
% si el archivo no fue encontrado, devuelve -1
% si el archivo es abierto para escritura y no existe, MATLAB lo crea.
% El archivo se cierra con fclose(numDeIdentificacion)

% el comando save escribe los cambios en disco.
```