# MEM522 SIGNAL PROCESSING
# LAB 1 INTRODUCTION TO MATLAB

## 1. General View

MATLAB is a high-level programming language that has been used extensively to solve complex engineering problems.

MATLAB works with three types of windows on your computer screen. These are the Command indow, the Figure window and the Editor window. The Figure window only pops up whenever you plot something. The Editor window is used for writing and editing MATLAB programs (called M-files) and can be invoked in Windows from the pulldown menu after selecting File | New | M-file.

The command window is the main window in which you communicate with the MATLAB interpreter. The MATLAB interpreter displays a command >> indicating that it is ready to accept commands from you.

You can use the command window as a calculator, or you can use it to call other MATLAB programs (M-files). Say you want to evaluate the expression *c* where a=1.2, b=2.3, c=4.5 and d=4. Then in the command window, type:

```
>> a = 1.2;
>> b=2.3;
>> c=4.5;
>> d=4;
>> a^3+sqrt(b*d)-4*c
ans = -13.2388
```

Note the semicolon after each variable assignment. If you omit the semicolon, then MATLAB echoes back on the screen the variable value.

## 2. Arithmetic Operations

There are four different arithmetic operators:

| | |
|---|---|
| + | Addition |
| − | Subtraction |
| * | Multiplication |
| / | Division (for matrices it also means inversion) |

There are also three other operators that operate on an element by element basis:

| | |
|---|---|
| .* | Multiplication of two vectors, element by element |
| ./ | Division of two vectors, element-wise |
| .^ | Raise all the elements of a vector to a power. |

```
>> X=[1,3,4]
>> Y=[4,5,6]
>> X+Y
ans= 5 8 10
```

For the vectors X and Y the operator + adds the elements of the vectors, one by one, assuming that the two vectors have the same dimension. In the above example, both vectors had the dimension 1 × 3, i.e., one row with three columns. An error will occur if you try to add a 1 × 3 vector to a 3 × 1 vector. The same applies for matrices.

To compute the dot product of two vectors, you can use the multiplication operator *. For the above example, it is:

```
>> X*Y’
ans = 43
```

Note the single quote after Y. The single quote denotes the transpose of a matrix or a vector. To compute an element by element multiplication of two vectors (or two arrays), you can use the .* operator:

```
>> X.*Y
ans = 4 15 24
```

That is, X.*Y means [1×4, 3×5, 4×6] = [4 15 24]. The ‘.*’ operator is used very often (and is highly recommended) because it is executed much faster compared to the code that uses for loops.

## 3. Complex Numbers

MATLAB also supports complex numbers. The imaginary number is denoted with the symbol i or j, assuming that you did not use these symbols anywhere in your program (that is very important!). Try the following:

```
>> z=3 + 4i
>> conj(z)      % computes the conjugate of z
>> angle(z)     % computes the phase of z
>> real(z)      % computes the real part of z
>> imag(z)      % computes the imaginary part of z
>> abs(z)       % computes the magnitude of z
```

You can also define the imaginary number with any other variables you like. Try the following:

```
>> img=sqrt(-1)
>> z=3+4*img
>> exp(pi*img)
```

## 4. Array Indexing

In MATLAB, all arrays (vectors) are indexed starting with 1, i.e., y(1) is the first element of the array y. Note that the arrays are indexed using parenthesis (.) and not square brackets [.] as in C/C++. To create an array having as elements the integers 1 through 6, just enter:

```
>> x=[1,2,3,4,5,6]
```

Alternatively, you can use the : notation,

```
>> x=1:6
```

The : notation above creates a vector starting from 1 to 6, in steps of 1. If you want to create a vector from 1 to 6 in steps of say 2, then type:

```
>> x=1:2:6
Ans = 1 3 5
```

Extracting or inserting numbers in a vector can be done very easily. To concatenate an array, you can use the [] operator, as shown in the example below:

```
>> x=[1:3 4 6 100:110]
```

To access a subset of the array, try the following:

```
>> x=(3:7)
>> length(x)             % gives the size of the array or vector
>> x=(2:2:length(x))
```

## 5. Allocating Memory

You can allocate memory for one-dimensional arrays (vectors) using the zeros command. The following command allocates memory for a 100-dimensional array:

```
>> Y=zeros(100,1);
>> Y(30)
ans = 0
```

Similarly, you can allocate memory for two-dimensional arrays (matrices). The command

```
>> Y=ones (4,5)
```

defines a 4 X 5 matrix. Similar to the zeros command, you can use the command ones to define a vector containing all ones,

```
>> Y=ones(1,5)
ans= 1 1 1 1 1
```

```
>> Y=rand (3,2)
```

defines a 7 X 8 matrix. Similar to the zeros command, you can use the command rand to define a vector containing random numbers < 1,

```
Y=

0.9597        0.2223

0.6543        0.4585

0.8614        0.5732
```

## 6. Special Characters and Functions

Some common special characters used in MATLAB are given below:

| Symbol | Meaning |
|---|---|
| **pi** | π(3.14...) |
| **sqrt** | indicates square root e.g., sqrt(4)=2 |
| ˆ | indicates power(e.g., 3ˆ2=9) |
| **abs** | Absolute value \| .\| e.g., abs(-3)=3 |

| | |
|---|---|
| **NaN** | Not-a-number, obtained when comparing mathematically undefined operations, such as 0/0 |
| **Inf** | Represents +∞ |
| **;** | Indicates the end of a row in a matrix. It is also used to suppress printing on the screen (echo off) |
| **%** | Denotes a comment. Anything to the right of % is ignored by the MATLAB interpreter and is considered as comments |
| **’** | Denotes transpose of a vector or matrix. It’s also used to define strings, e.g.,str1=’DSP’; |

Some special functions are given below:

| | |
|---|---|
| **length(x)** | gives the dimension of the array x |
| **find** | Finds indices of nonzero elements. |

```
>> x=1:10;
>> length(x)
ans = 10
```

The function find returns the indices of the vector X that are non-zero. For example, **I= find(X>100),** finds all the indices of X when X is greater than 100. So for the above

```
>> find(x> 4)

ans = 5 6 7 8 9 10
```

## 7. Plotting

You can plot arrays using MATLAB’s function plot. The function plot(.) is used to generate line plots. The function stem(.) is used to generate “picket-fence” type of plots.

```
>> x=1:20;

>> plot(x)

>> stem(x)
```

More generally, plot(X,Y) plots vector Y versus vector X. Various line types, plot symbols and colors may be obtained using plot(X,Y,S) where S is a character string indicating the color of the line, and the type of line (e.g., dashed, solid, dotted, etc.). Examples for the string S include:

| | |
|---|---|
| **r** | Red |
| **+** | Plus |
| **--** | Dashed |
| **g** | Green |
| ***** | Star |
| **B** | Blue |

**S**        Square

You can insert x-labels, y-labels and title to the plots, using the functions xlabel(.), ylabel(.) and title(.) respectively. To plot two or more graphs on the same figure, use the command subplot. For instance, to show the above two plots in the same figure, type:

**>> subplot(2,1,1), plot(x)**

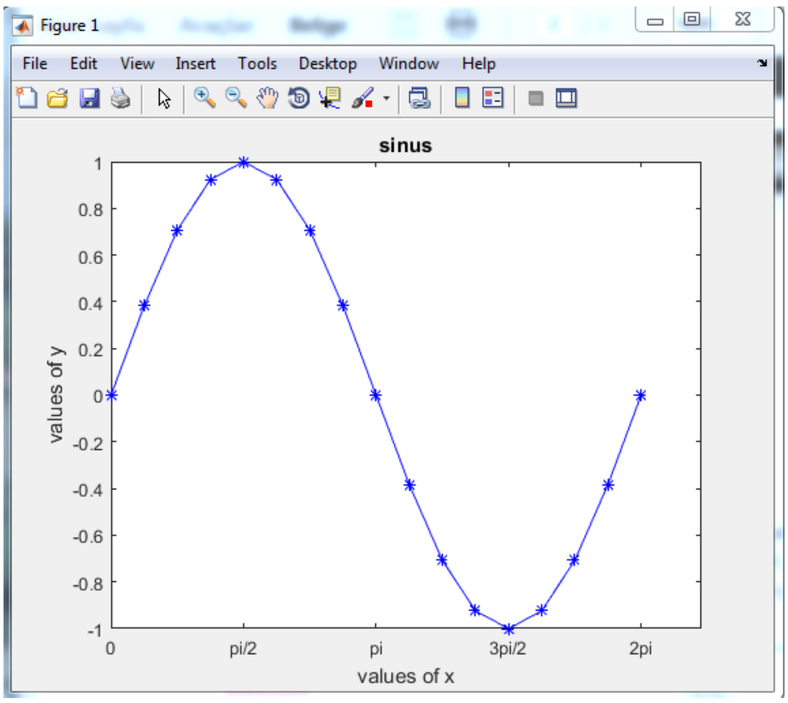
**>> subplot(2,1,2), stem(x)**

The (m,n,p) argument in the subplot command indicates that the figure will be split in m rows and n columns. The ‘p’ argument takes the values 1, 2 . . . m×n. In the example above, m = 2, n = 1,and, p = 1 for the top figure and p = 2 for the bottom figure.

**** To get more help on plotting, type: help plot or help subplot.

## 8. Sin, Cos etc.

```
clc
clear all
x=[0: pi/8 : 2*pi];
y=sin(x);
plot(x,y,'b-*')
title('sinus')
xlabel('values of x')
ylabel('values of y')
set(gca,'Xtick', 0: pi/2 : 2*pi)
set(gca,'XTickLabel', {'0','pi/2','pi','3pi/2','2pi'})
```



## 9. Excercises

1. Consider the discrete-time signal $x_M[n] = \sin(2\pi .M.n/N)$ and assume N=10. For M=3,5,6,7,8 ,10, plot $x_M[n]$ on the interval $0 \le n \le 2N - 1$. Use **stem** to create your plots, and be sure to appropriately **label your axes**.

2. Now consider the following signals:

   $X_1[n] = 2. \sin(2\pi n/N) + \sin(3\pi n/N)$

   $X_2[n] = \cos(2\pi n/N) + 3.\cos(5\pi n/2N)$

   Assume N=8 for each signal. **Determine whether or not each signal periodic**. Plot the signal for $0 \le n \le 7N$. Remember to use **stem** and to **appropriately label your axes**