Communication Systems

Lab # 02

Compiled by: Engr. Mariam Shafqat

Date of conduction: 23 and 24 September 2010

Date of Submission: 29 and 30 September 2010

Venue: Computer Engineering Lab ; CPED, UET Taxila

Introduction to Data Types:

The Logical data Type

This data type can have only one of the two values true or false;they are declared as:

```
>> a1=true
a1 =
    1
>> a2=false
a2 =
    0
```

Relational Operators

These are the operators that yield a logical result; the format is

A1 op a2

The relational operators are

1. ==
2. ~=
3. <
4. <=
5. >
6. >=

```
>> 3==4
ans =
    0
>> 3~=4
ans =
    1
>> 3<4
ans =
    1
>> 3<=4
ans =
    1
>> 3>4
ans =
    0
```

```
>> 3>=4
ans =
    0
```

They are also used to compare scalar values with an array

```
>> a=[1 2;3 4];
>> b=1;
>> a<b
ans =
     0     0
     0     0
```

This result is similar to
False false
False false

The relational equivalence operator == is different from = operator. The first one is relational operator and second one is the assignment operator.

Try this one out:

```
>> a=0;
>> b=sin(pi);
>> a==b
ans =
    0
```

Why the answer is 0; it should be one;
On the other hand

```
>> a=0;
>> b=sin(pi);
>> a==round(b)
ans =
    1
```

Logic Operators:

These are the operators with two or more logical operators that yield a logical result.the format of the operation is

Li op l2

These operators are of three types

Logical AND (& and &&)
Logical OR (| and ||)
Exclusive OR (xor)

Logical AND

This operator return 1 if and only if both operands are 1;
The difference between & and && is that && support short circuit evaluation. If one of the operand is not 1 it will immediately return 0; it will not bother to evaluate the rest of the expression. But & first evaluate the whole expression and then return the answer.
The second difference is that; the && work only with scalar values but & work with both the scalar and array values, as long as the array dimensions are same.

Logical OR

The result of this operator is true if either of the operand is true. Again there are two operators that have the same differences as & and &&.

Logical NOT

Turns false to true and true to false.

Using Numerical Data with logic operators:

Simply MATLAB considers all non-zero numerical data as true and zero data as false.

Hierarchy of operations

Following is the hierarchy of operations:

1. All arithmetic operations to be evaluated from left to right
2. All relational operators
3. All ~ operators are evaluated
4. All & and && are evaluated
5. All | and || and xor are evaluated

Logical Functions

These functions return true when the question being evaluated is true else false.
Following are some of these functions:

1. Ischar()
2. Isempty()
3. Isinf()
4. Isnan()
5. Isnumeric()
6. Logical()

Branches

The if construct

```
if   control_expression1
        Statement1
        Statement2
        Statement3
else if Control_expression
        Statement1
        Statement2
        Statement3
else
```

```
        Statement1
        Statement2
        Statement3
end
```

```
>> x=3;
>> if x<2
disp('Number is less than 2');
else
disp('Number is greater than 2');
end
Number is greater than 2
```

The switch statement

The format is

```
switch (switch_expression)
case case_expression1
        Statement1
        Statement2
case case_expression2
        Statement1
        Statement2
case case_expression3
        Statement1
        Statement2
end
```

```
>> a=input('plz enter the value of the number')
plz enter the value of the number  1
a =
1
>> switch(a)
case 1
disp('the inputted value is 1');
case 2
disp('the inputted value is 2');
case 3
disp('the inputted value is 3');
end
the inputted value is 1
```

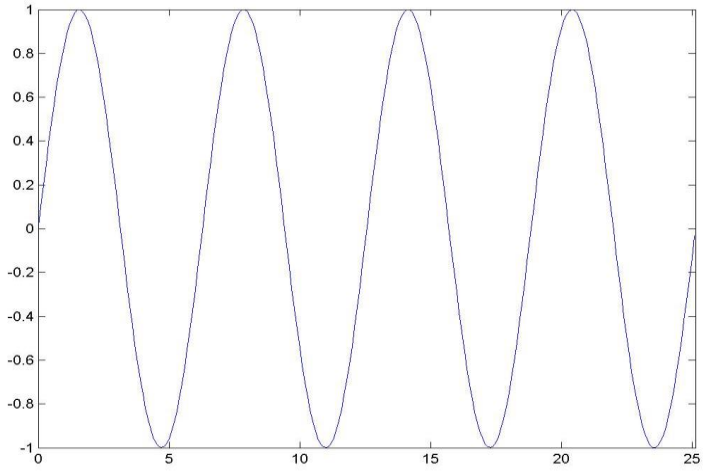
The try/Catch Construct

This statement is used to find errors form the program. If an error occurs in try block; the program will automatically shift to catch block and if an error occurs in catch block; it will automatically shift to try block. Means that's program will never halt in any case.

```
>> a=[1 -3 2 5];
>> try
switch a
case 2
disp('elemnet i s2');
case 4
disp('element is 4');
end
catch
disp('element is either 1 -3 2 5')
end
element is either 1 -3 2 5
```

Additional Plotting features

```
>> x=0:0.1:8*pi;
>> y=sin(x);
>> plot(x,y)
>> axis([0 8*pi -1 1])
>> axis equal
>> axis square
>> axis normal
>> axis off
>> axis on
```



Plotting multiple plots on the same area

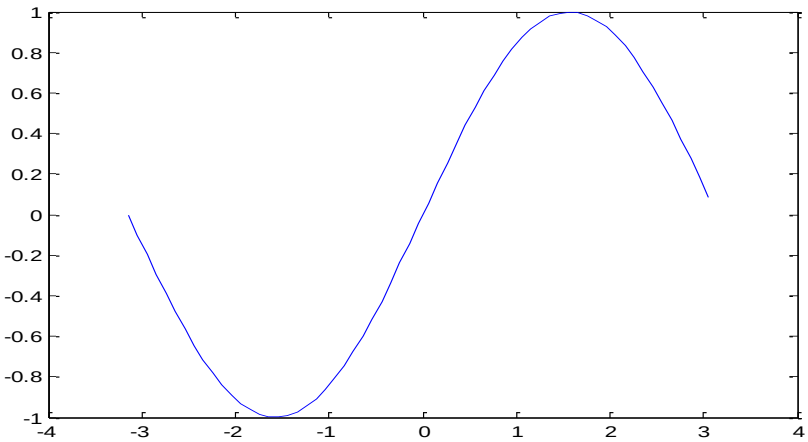
Normally when multiple plots are plotted on an area; the previous plots get deleted. To avoid this situation; use "hold" command; the hold on command will display all the previous plots on the same area. To deactivate this facility, use hold off command.

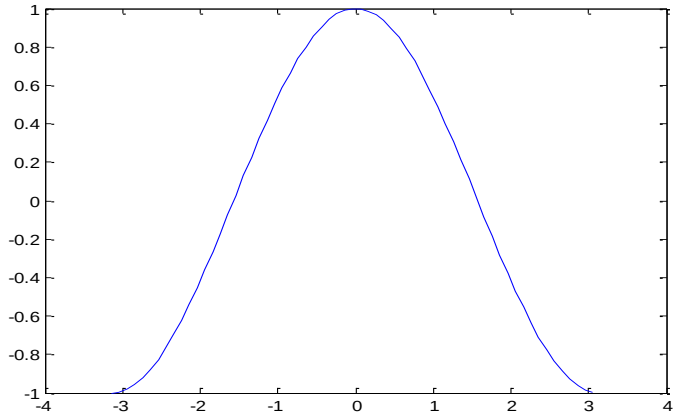
```
>> x=-pi:0.1:pi;
>> y=sin(x);
>> z=cos(x);
>> plot(x,y);
>> hold on;
>> plot(x,z);
>> legend('sin','cos','TL');
```

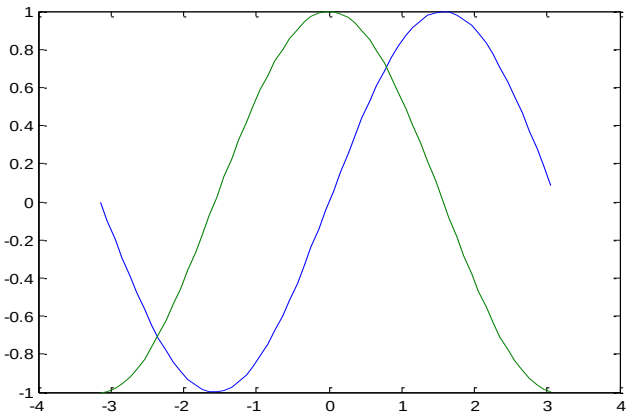
Creating Multiple plots

Done with the help of "figure " command. Each figure is identified with a subscript like figure(1),figure(2) e.t.c.

```
>> figure(1);
>> x=-pi:0.1:pi;
>> y=sin(x);
>> plot(x,y);
>> figure(2);
>> z=cos(x);
>> plot(x,z);
>> figure(3);
>> plot(x,y,x,z);
```



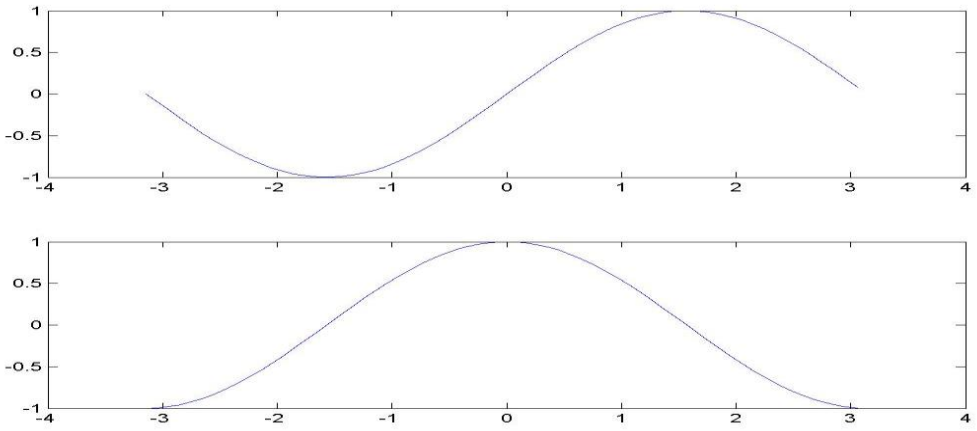




Subplots:

Done with the command subplot(m,n,p). this command divides the current figure into mXn equal-sized regions, arranged into m rows and n columns, and creates a set of axes at position p to receive all current plotting commands.

```
>> figure(1);
>> subplot(2,1,1);
>> x=-pi:0.1:pi;
>> y=sin(x);
>> plot(x,y);
>> subplot(2,1,2);
>> z=cos(x);
>> plot(x,z);
```



Enhanced control of plotted lines:
Plot(x,y,'PropertName',value,……)

The 'for 'loop

For index=expression
Statement1
Statement2
.
.
.
.end

Lab Tasks:

Question # 01:

Try all the commands and show the output during lab session.

Question # 02:

Implement and plot unit impulse function.

Question # 03:

Implement and plot unit step function.

Question # 04:

Calculate factorial of a number.

Question # 05:

Implement and plot using all the plotting commands for the equation $y(x)=x^2-3x+2$ *for all the values of x between -1 and 3 with the increment of 0.1.*

****