

Scilab Manual for  
Signals and Systems  
by Prof Priyen S. Patel  
Electrical Engineering  
Swarnim Startup & Innovation University<sup>1</sup>

Solutions provided by  
Prof Priyen S. Patel  
Electrical Engineering  
Swarnim Startup & Innovation University

March 10, 2025

<sup>1</sup>Funded by a grant from the National Mission on Education through ICT, <http://spoken-tutorial.org/NMEICT-Intro>. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website <http://scilab.in>



# Contents

List of Scilab Solutions	4
1 Develop a program to generate Following Continuous Signal a) Sinusoidal; b)Cosine; c)Triangle; d)Square Wave.	6
2 Develop a program to generate Following Discrete Signal a) Impulse; b)Step; c)Ramp; d)Exponential.	9
3 Develop a program for addition of two continuous signals using Scilab.	14
4 Develop a program for addition of two discrete signals using Scilab.	17
5 Develop a program for Aliasing Process using Scilab.	20
6 Develop a program for Linear Convolution of two sequences using Scilab.	24
7 Develop a program for Circular Convolution of two sequences using Scilab.	27
8 Develop a program to perform cross correlation operation using Scilab.	30
9 Develop a program to perform Auto correlation operation using Scilab.	33

10	Develop a program to obtain Z transform of basic function using Scilab.	36
11	Develop a program to obtain pole zero plot of given transfer function using Scilab.	39
12	Develop a program to understand the concept of amplitude modulation.	42
13	Develop a program to understand the concept of frequency modulation.	45

# List of Experiments

Solution 1.01	Continuous Signal . . . . .	6
Solution 2.01	Discrete wave with positive exponential . . . . .	9
Solution 2.02	Discrete waves with negative exponential . . . . .	11
Solution 3.01	Addition of Continuous signal . . . . .	14
Solution 4.01	Addition of Discrete signal . . . . .	17
Solution 5.01	Aliasing Effect . . . . .	20
Solution 6.01	Linear Convolution . . . . .	24
Solution 7.01	Circular Convolution . . . . .	27
Solution 8.01	Cross correlation . . . . .	30
Solution 9.01	Auto correlation . . . . .	33
Solution 10.01	Z transform . . . . .	36
Solution 11.01	Pole Zero form Transfer function . . . . .	39
Solution 12.01	Amplitude Modulation . . . . .	42
Solution 13.01	Frequency Modulation . . . . .	45

# List of Figures

1.1	Continuous Signal . . . . .	8
2.1	Discrete wave with positive exponential . . . . .	11
2.2	Discrete waves with negative exponential . . . . .	13
3.1	Addition of Continuous signal . . . . .	15
4.1	Addition of Discrete signal . . . . .	18
5.1	Aliasing Effect . . . . .	21
6.1	Linear Convolution . . . . .	25
7.1	Circular Convolution . . . . .	28
8.1	Cross correlation . . . . .	31
9.1	Auto correlation . . . . .	34
10.1	Z transform . . . . .	38
11.1	Pole Zero form Transfer function . . . . .	40
11.2	Pole Zero form Transfer function . . . . .	41
12.1	Amplitude Modulation . . . . .	43
13.1	Frequency Modulation . . . . .	46

## Experiment: 1

Develop a program to generate Following Continuous Signal a) Sinusoidal; b) Cosine; c) Triangle; d) Square Wave.

Scilab code Solution 1.01 Continuous Signal

```
1 //Experiment-1
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM: Develop a program to generate Following
   Continuous Signal a) Sinusoidal; b) Cosine; c)
   Triangle; d) Square Wave.
7
8 clear all
9 clc
10 V = input('Enter the value of Voltage in volts : ')
      // Example v= 20 Volt
11 f = input('Enter the value of frequency in Hertz : ')
      // Example f=50 Hz
```

```

12 t=0:1/(1000*f):5/f;
13 // Generation of Sine Wave.
14 y1=V*(sin(2*%pi*f*t));
15 xgrid;
16 subplot(221)
17 plot(t,y1)
18 xlabel('Time (sec) ')
19 ylabel('y = v*sinwt')
20 title('Sine wave','fontsize',4)
21 // Generation of Cos Wave.
22 y2=V*(cos(2*%pi*f*t));
23 subplot(222)
24 plot(t,y2)
25 xgrid;
26 xlabel('Time (sec) ')
27 ylabel('y = v*coswt')
28 title('Cosine wave','fontsize',4)
29 // Generation of Triangle Wave.
30 t1 =0:( %pi /4) :(4* %pi ) ;
31 y3 = V *sin (2* t1) ;
32 a = gca () ;
33 subplot(223)
34 plot (t1 ,y3 ) ;
35 xgrid;
36 xlabel('Time (sec) ')
37 ylabel('Amplitude')
38 title('Triangle wave','fontsize',4)
39 // Generation of Square Wave.
40 t3=0:1/(1000*f):0.6;
41 y4=(V-1)*squarewave(2*%pi*10*t3);
42 subplot(224)
43 plot(t3,y4)
44 xgrid;
45 xlabel('Time (sec) ')
46 ylabel('Amplitude')
47 title('Square wave','fontsize',4)
48
49 //... Consol Window entry...

```



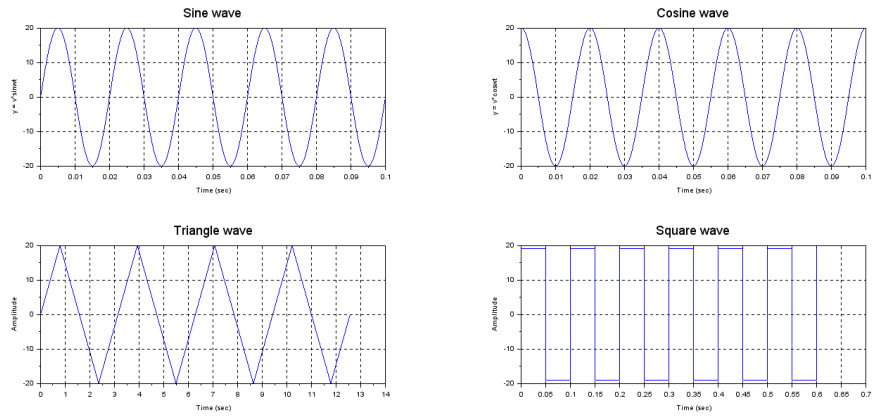


Figure 1.1: Continuous Signal

```

50 //Enter the value of Voltage in volts : 20
51 //Enter the value of frequency in Hertz : 50

```

---

## Experiment: 2

Develop a program to generate  
Following Discrete Signal a)  
Impulse; b)Step; c)Ramp;  
d)Exponential.

Scilab code Solution 2.01 Discrete wave with positive exponential

```
1 //Experiment-2
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM:  Devlop a program to generate Following
      Discrete Signal a) Impulse; b)Step; c)Ramp; d)
      Exponential.
7
8 clc ;
9 clf ;
10 clear all;
11 L =input("Enter the Length of signal="); // L=10 (
      must be greater than 2)
12 b =input("Enter the value of Exponential co-
```

```

    efficient=""); // b = 0.5
13 // positive value for incremental Exponential signal
14
15 n=-L:L;
16 //Generation of Unit Impulse Signal
17 x1=[zeros(1,L),ones(1,1),zeros(1,L)];
18 a= gca ();
19 a. y_location = "middle"
20 subplot(221)
21 plot2d3 (n,x1);
22 title ( 'Unit Impulse Signal', "fontsize", 3 );
23 xlabel ( ' Number of Sample————>' );
24 ylabel ( 'Amplitude ——>' );
25 //Generation of Unit Step Signal
26 x2=[zeros(1,L),ones(1,L +1)];
27 a= gca ();
28 a. y_location = "middle";
29 subplot(222)
30 plot2d3 (n,x2);
31 title ( 'Unit Step', "fontsize", 3 );
32 xlabel ( ' Number of Sample————>' );
33 ylabel ( 'Amplitude ——>' );
34 //Generation of Ramp Signal
35 x3=[zeros(1,L),0:L];
36 a = gca ();
37 a. y_location = 'middle' ;
38 subplot(223)
39 plot2d3 (n,x3);
40 title ( 'Ramp of signal', "fontsize", 3);
41 xlabel ( ' Number of Sample————>' );
42 ylabel ( 'Amplitude ——>' );
43
44 //Generation of Exponential Signal
45 t = -2:0.1:2;
46 x4= exp (b*t);
47 subplot(224)
48 plot2d3 (x4);
49 title ( 'Exponential Signal', "fontsize", 3 );

```

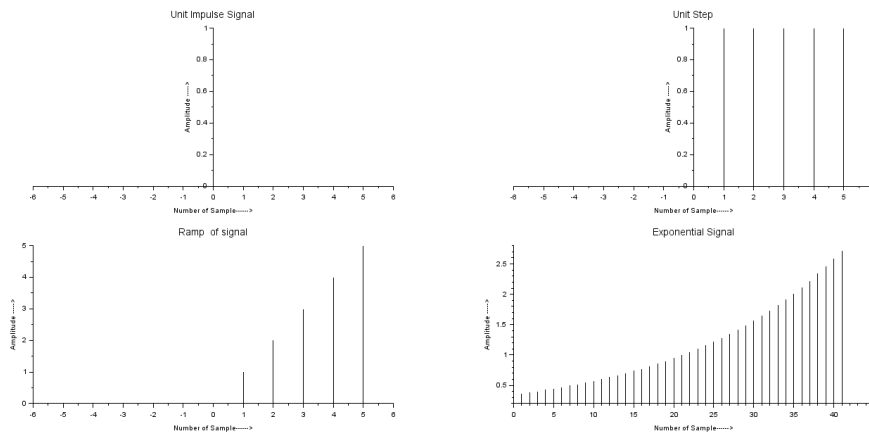


Figure 2.1: Discrete wave with positive exponential

```

50 xlabel ( ' Number of Sample————>' );
51 ylabel ( ' Amplitude —————>' );
52
53 // ... Console input
54 //Enter the Length of signal=5
55
56 //Enter the value of Exponential co-efficient=0.5

```

---

### Scilab code Solution 2.02 Discrete waves with negative exponential

```

1 //Experiment-2
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM: Develop a program to generate Following
  Discrete Signal a) Impulse; b) Step; c) Ramp; d)
  Exponential.
7

```

```

8  clc ;
9  clf ;
10 clear all;
11 L =input("Enter the Length of signal="); // L=10 (
    must be greater than 2)
12 b =input("Enter the value of Exponential co-
    efficient="); // b = 0.5
13 n=-L:L;
14 //Generation of Unit Impulse Signal
15 x1=[zeros(1,L),ones(1,1),zeros(1,L)];
16 a= gca ();
17 a. y_location = "middle"
18 subplot(221)
19 plot2d3 (n,x1);
20 title ( 'Unit Impulse Signal', "fontsize",3 );
21 xlabel ( ' Number of Sample————>' );
22 ylabel ( 'Amplitude ——>' );
23 //Generation of Unit Step Signal
24 x2=[zeros(1,L),ones(1,L +1)];
25 a= gca ();
26 a. y_location = "middle";
27 subplot(222)
28 plot2d3 (n,x2);
29 title ( 'Unit Step', "fontsize",3 );
30 xlabel ( ' Number of Sample————>' );
31 ylabel ( 'Amplitude ——>' );
32 //Generation of Ramp Signal
33 x3=[zeros(1,L),0:L];
34 a = gca ();
35 a. y_location = 'middle' ;
36 subplot(223)
37 plot2d3 (n,x3);
38 title ( 'Ramp of signal', "fontsize",3);
39 xlabel ( ' Number of Sample————>' );
40 ylabel ( 'Amplitude ——>' );
41
42 //Generation of Negative value Exponential Signal
43 t = -2:0.1:2;

```

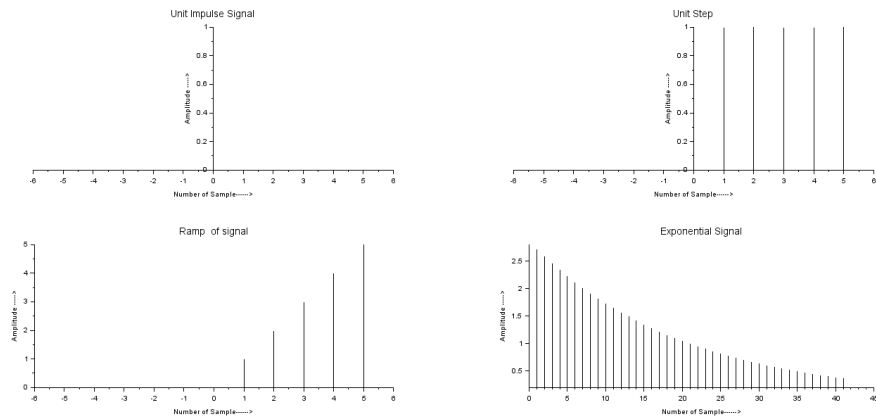


Figure 2.2: Discrete waves with negative exponential

```

44 x4= exp (-b*t);
45 subplot(224)
46 plot2d3 (x4);
47 title ( 'Exponential Signal ', "fontsize", 3 );
48 xlabel ( ' Number of Sample————>' );
49 ylabel ( ' Amplitude —————>' );
50
51 // ... Console input
52 //Enter the Length of signal=5
53
54 //Enter the value of Exponential co-efficient=0.5

```

---

## Experiment: 3

Develop a program for addition of two continuous signals using Scilab.

Scilab code Solution 3.01 Addition of Continuous signal

```
1 //Experiment-3
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM: Develop a program for addition of two
   Continuous signals using Scilab.
7
8
9 // Addition of Continuous Signals
10 clc ;
11 clear all;
12 V = input('Enter the value of Voltage in volts : ')
   // Example v= 5 Volt
13 f = input('Enter the value of Signal 1 frequency in
```

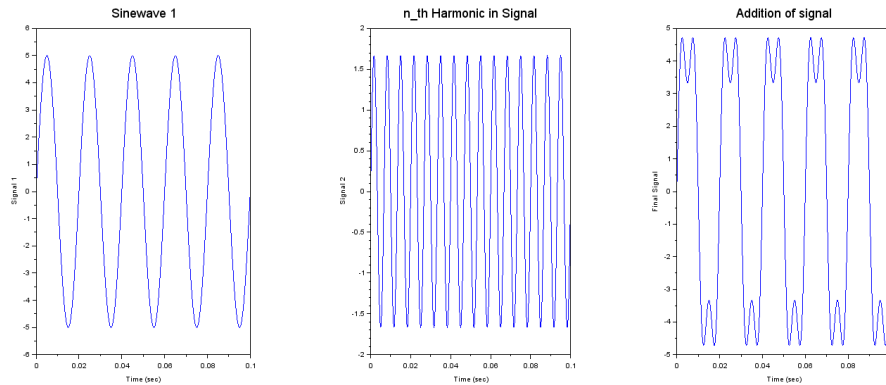


Figure 3.1: Addition of Continuous signal

```

Hertz : ') // Example f1=50 Hz
14 n = input('Enter the number of harmonic in signal :
           ') // Example n= 3
15 t=0:1/(1000*f):5/f;
16 y1=V*(sin(2*pi*f*t));
17 y2=(V/n)*(sin(2*pi*(n*f)*t));
18 y3=y1+y2;
19 subplot (1 ,3 ,1);
20 plot (t,y1);
21 xlabel('Time (sec) ');
22 ylabel('Signal 1')
23 title('Sinewave 1','fontsize',4)
24 subplot (1 ,3 ,2);
25 plot (t,y2);
26 xlabel('Time (sec) ');
27 ylabel('Signal 2')
28 title('n_th Harmonic in Signal','fontsize',4)
29 subplot (1 ,3 ,3);
30 plot (t,y3);
31 title('Addition of signal','fontsize',4)
32 xlabel('Time (sec) ');
33 ylabel('Final Signal')
34

```



```
35 // ..... Execution Consol Value
    .....//
36 //Enter the value of Voltage in volts : 5
37 //Enter the value of Signal 1 frequency in Hertz :
    50
38 //Enter the number of harmonic in signal : 3
```

---

## Experiment: 4

Develop a program for addition of two discrete signals using Scilab.

Scilab code Solution 4.01 Addition of Discrete signal

```
1 //Experiment-3
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM: Develop a program for addition of two
   discrete signals using Scilab.
7
8
9 // Addition of Discontinuous Signals
10
11 clc ;
12 clear all;
13
14 L =input("Enter the Length of signal="); // L=10 (
```

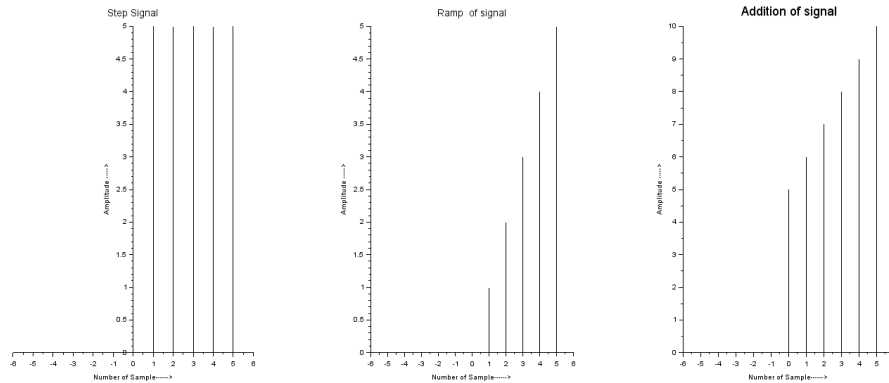


Figure 4.1: Addition of Discrete signal

```

    must be greater than 2)
15 n=-L:L;
16 //Generation of Step Signal
17 x1=5*[zeros(1,L),ones(1,L +1)];
18 a= gca ();
19 a. y_location = "middle";
20 subplot(131)
21 plot2d3 (n,x1);
22 title ( 'Step Signal', "fontsize",3 );
23 xlabel ( ' Number of Sample————>' );
24 ylabel ( 'Amplitude ——>' );
25 //Generation of Ramp Signal
26 x2=[zeros(1,L),0:L];
27 a = gca ();
28 a. y_location = 'middle' ;
29 subplot(132)
30 plot2d3 (n,x2);
31 title ( 'Ramp of signal', "fontsize",3);
32 xlabel ( ' Number of Sample————>' );
33 ylabel ( 'Amplitude ——>' );
34 x3=x1+x2; // Addition of two signals
35 subplot(133)
36 plot2d3 (n,x3);

```

```
37 title('Addition of signal ','fontsize',4)
38 xlabel(' Number of Sample————>')
39 ylabel('Amplitude ———>')
40
41 // ..... Execution Consol Value
   .....//
42 // Enter the Length of signal=5
```

---

## Experiment: 5

# Develop a program for Aliasing Process using Scilab.

Scilab code Solution 5.01 Aliasing Effect

```
1 //Experiment-5
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM: Develop a program for ALIASING Process using
   scilab
7
8 clc ;
9
10 clear all;
11
12 f=input("Enter the frequency of continuous signal=")
   ;// f= 1000
13 v=input("Enter the Amplitude of continuous signal=")
   ; // v = 10
14 fs=input (" Enter the sampling Frequency Fs =") ; //
```

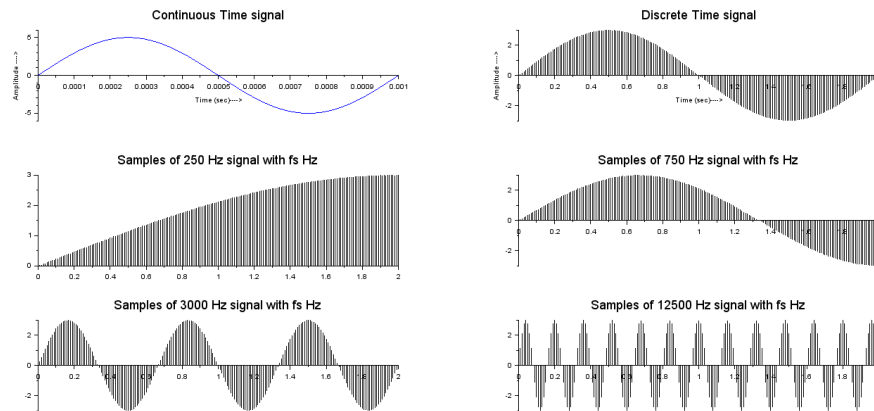


Figure 5.1: Aliasing Effect

```

fs= 2000Hz
15 t=0:0.00001:1/f;
16 x1=v*sin(2*%pi*f*t);
17 subplot(3,2,1)
18 a=gca();
19 a.x_location="origin";
20 a.y_location="origin";
21 plot(t,x1)
22 xlabel('Time (sec)——>')
23 ylabel('Amplitude ——>')
24 title('Continuous Time signal','fontsize',4)
25 f2=f/fs;
26 n=0:0.01:1/f2;
27 y1=3*sin(2*%pi*f2*n);
28 subplot(3,2,2)
29 a = gca ();
30 a.x_location="origin";
31 a.y_location="origin";
32 plot2d3("gmn",n,y1)
33 xlabel('Time (sec)——>')
34 ylabel('Amplitude ——>')
35 title('Discrete Time signal','fontsize',4)
36 x2 = 250;

```

```

37 y2=3*sin(2*pi*(x2/fs)*n) ;
38 subplot (3,2,3)
39 a = gca () ;
40 a. x_location ="origin";
41 a. y_location = "origin";
42 plot2d3 ("gnn",n,y2)
43 title ('Samples of 250 Hz signal with fs Hz',"
        fontsize",4)
44 x3 = 750;
45 y3 = 3*sin(2*pi*(x3/fs)*n) ;
46 subplot (3,2,4)
47 a = gca () ;
48 a. x_location ="origin";
49 a. y_location = "origin";
50 plot2d3 ("gnn",n,y3)
51 title ('Samples of 750 Hz signal with fs Hz',"
        fontsize",4)
52 x4 = 3000;
53 y4 = 3*sin(2*pi*(x4/fs)*n) ;
54 subplot (3,2,5)
55 a = gca () ;
56 a. x_location ="origin";
57 a. y_location = "origin";
58 plot2d3 ("gnn",n,y4)
59 title ('Samples of 3000 Hz signal with fs Hz',"
        fontsize",4)
60 x5 = 12500;
61 y5 = 3*sin(2*pi*(x5/fs)*n) ;
62 subplot (3,2,6)
63 a = gca () ;
64 a. x_location ="origin";
65 a. y_location = "origin";
66 plot2d3 ("gnn",n,y5)
67 title ('Samples of 12500 Hz signal with fs Hz',"
        fontsize",4)
68 //Example
69 // E n t e r t h e f r e q u e n c y o f C o n t i n
    u o u s T i m e S i g n a l :=1000

```

```
70 // Enter the Sampling frequency Fs=  
2000
```

---



## Experiment: 6

Develop a program for Linear Convolution of two sequences using Scilab.

Scilab code Solution 6.01 Linear Convolution

```
1 //Experiment-6
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM:  Develop a program for Linear Convolution of
       two sequences using scilab
7 clc ;
8 clear ;
9 x = input ("ENTER THE FIRST SEQUENCE [use square
           breackt- arrey form]= "); // x ( n ) =[1 2 3 4 ]
10 h = input ("ENTER THE SECOND SEQUENCE ="); //h ( n )
           =[1 2 3 ]
11 l=length (x)+length (h)-1;
12 y = convol(x,h)
```

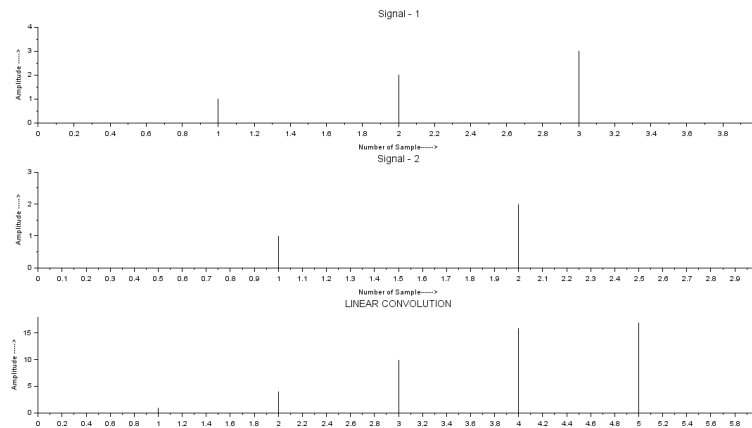


Figure 6.1: Linear Convolution

```

13 disp(y, 'Linear convolution is =')
14 subplot (311)
15 plot2d3('gmn',x);
16 a = gca ();
17 a.x_location = "origin";
18 a.y_location = "origin";
19 title ( 'Signal - 1',"fontsize",3 );
20 xlabel ( ' Number of Sample————>' );
21 ylabel ( ' Amplitude ——>' );
22 subplot (312)
23 plot2d3(h);
24 a = gca ();
25 a.x_location = "origin";
26 a.y_location = "origin";
27 title ( 'Signal - 2',"fontsize",3 );
28 xlabel ( ' Number of Sample————>' );
29 ylabel ( ' Amplitude ——>' );
30 subplot (313)
31 plot2d3(y);
32 a = gca ();
33 a.x_location = "origin";
34 a.y_location = "origin";
35 title ( 'LINEAR CONVOLUTION',"fontsize",3 );

```

```

36 xlabel ( ' Number of Sample————>' );
37 ylabel ( 'Amplitude ——>' );
38
39 //.....Execution in scilab console....
40
41 //ENTER THE FIRST SEQUENCE [use square breackt-
    array form]= [1 2 3 4]
42 //ENTER THE SECOND SEQUENCE =[1 2 3]
43 //—————Answer of linear convolution—————
44 //Linear convolution is =
45
46 //      1.      4.      10.      16.      17.      12.

```

---

## Experiment: 7

Develop a program for Circular Convolution of two sequences using Scilab.

Scilab code Solution 7.01 Circular Convolution

```
1 //Experiment-7
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM:  Develop a program for circular Convolution of
       two sequences using scilab
7 clc ;
8 clf();
9 clear ;
10 x = input ("ENTER THE FIRST SEQUENCE [use square
           bracket- array form]= "); // x ( n ) =[1 2 3 5 6
           4]
11 h = input ("ENTER THE SECOND SEQUENCE ="); //h ( n )
           =[1 2 3 1]
```

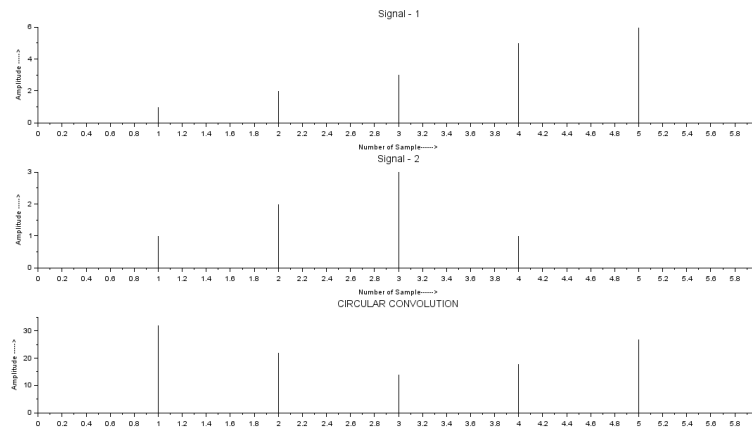


Figure 7.1: Circular Convolution

```

12 N1= length(x);
13 N2= length(h);
14 N= max (N1 ,N2);
15 N3=N1-N2;
16 if(N3>=0) then
17     h=[h,zeros(1,N3)];
18 else
19     x=[x,zeros(1,-N3)];
20 end
21 for n =1: N
22     y(n)=0;
23 for i =1:N
24     j=n-i+1;
25     if(j<=0)
26         j=N+j;
27     end
28     y(n)=y(n)+x(i)*h(j)
29 end
30 end
31 disp(y,'Circular convolution is =')
32 subplot (311)
33 plot2d3('gmn',x);
34 a = gca ();

```

```

35 a.x_location = "origin";
36 a.y_location = "origin";
37 title ( 'Signal - 1',"fontsize",3 );
38 xlabel ( ' Number of Sample————>' );
39 ylabel ( 'Amplitude ——>' );
40 subplot (312)
41 plot2d3('gmn',h);
42 a = gca ();
43 a.x_location = "origin";
44 a.y_location = "origin";
45 title ( 'Signal - 2',"fontsize",3 );
46 xlabel ( ' Number of Sample————>' );
47 ylabel ( 'Amplitude ——>' );
48 subplot (313)
49 plot2d3('gmn',y);
50 a = gca ();
51 a.x_location = "origin";
52 a.y_location = "origin";
53 title ( 'CIRCULAR CONVOLUTION',"fontsize",3 );
54 xlabel ( ' Number of Sample————>' );
55 ylabel ( 'Amplitude ——>' );
56
57
58 //.... Execution in Scilab 5.4.1 console....
59
60 //ENTER THE FIRST SEQUENCE [use square brack-
    array form]= [1 2 3 5 6 4]
61 //ENTER THE SECOND SEQUENCE =[1 2 3 1]
62
63 // Circular convolution is =
64
65 // 32 22 14 18 27 34

```

---

## Experiment: 8

Develop a program to perform cross correlation operation using Scilab.

Scilab code Solution 8.01 Cross correlation

```
1 //Experiment-8
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM:  Develop a program for Performing cross
      correlation operation using SCILAB code
7
8
9 clear ;
10 clc;
11 x1 =input ("ENTER THE FIRST SEQUENCE 01 [use square
      bracket- array form]= ");//[1 2 1 1]
12 x2 =input ("ENTER THE FIRST SEQUENCE 02 [use square
      bracket- array form]= "); //[1 1 2 1]
```

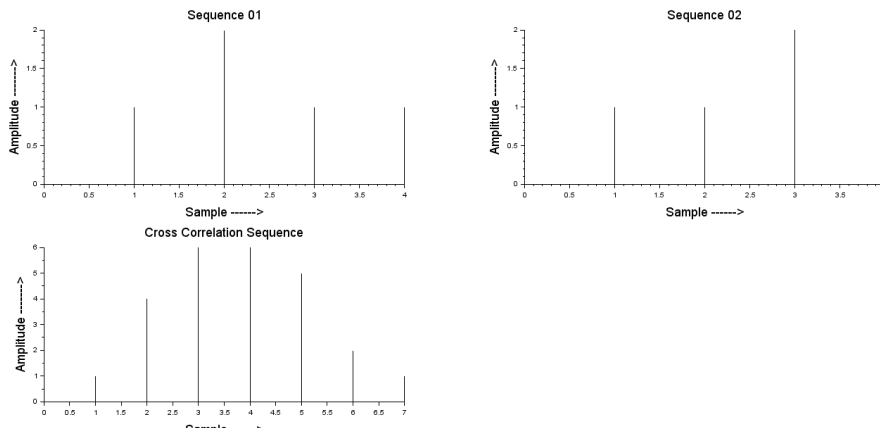


Figure 8.1: Cross correlation

```

13 n1= input ("ENTER THE Arrey length of sequence 2
            from starting point [use square bracket- array
            form]= "); //[1 2 3 4]
14 n2= input ("ENTER THE Arrey length of sequence 2
            from starting point [use square bracket- array
            form]= "); //[1 2 3 4]
15 subplot (2 ,2 ,1)
16 plot2d3 ('gmn ',n1,x1);
17 a = gca ();
18 a.x_location = "origin";
19 a.y_location = "origin";
20 xlabel ("Sample ----->" ," fontsize" ,4);
21 ylabel ("Amplitude ----->" ," fontsize" ,4);
22 title ("Sequence 01" ," fontsize" ,4);
23 subplot (2 ,2 ,2)
24 plot2d3 ('gmn ',n2,x2);
25 a = gca ();
26 a.x_location = "origin";
27 a.y_location = "origin";
28 xlabel ("Sample ----->" ," fontsize" ,4);
29 ylabel ("Amplitude ----->" ," fontsize" ,4);
30 title ("Sequence 02" ," fontsize" ,4);
31 [c, ind ]= xcorr (x1 ,x2) // function of cross

```



```

    correlation
32 [ind',c']
33 disp(c,'Cross Correlation Sequence is =')
34 subplot (2 ,2 ,3)
35 plot2d3 ('gmn',c)
36 a = gca ();
37 a.x_location = "origin";
38 a.y_location = "origin";
39 xlabel ("Sample ——>" ,"fontsize" ,4);
40 ylabel ("Amplitude ——>" ,"fontsize" ,4);
41 title ("Cross Correlation Sequence" ,"fontsize" ,4);
42
43 // ..... Execution in Consol .....
44
45 //ENTER THE FIRST SEQUENCE 01 [use square breackt-
    array form]=[1 2 1 1]
46 //ENTER THE FIRST SEQUENCE 02 [use square breackt-
    array form]=[1 1 2 1]
47 //ENTER THE Arrey length of sequence 2 from starting
    point [use square breackt- array form]=[1 2 3
    4]
48 //ENTER THE Arrey length of sequence 2 from starting
    point [use square breackt- array form]=[1 2 3
    4]
49
50 // Cross Correlation Sequence is =
51
52 //      1.      4.      6.      6.      5.      2.      1.

```

---

## Experiment: 9

Develop a program to perform  
Auto correlation operation  
using Scilab.

Scilab code Solution 9.01 Auto correlation

```
1 //Experiment-9
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM:  Develop a program for Performing Auto
      correlation operation using SCILAB code
7
8
9 clear ;
10 clc;
11 x1 =input ("ENTER THE FIRST SEQUENCE 01 [use square
      bracket- array form]= ");//[2 -1 2 3 1]
12 n = length (x1);
13 s1= input ("ENTER THE starting point of array = ");
```

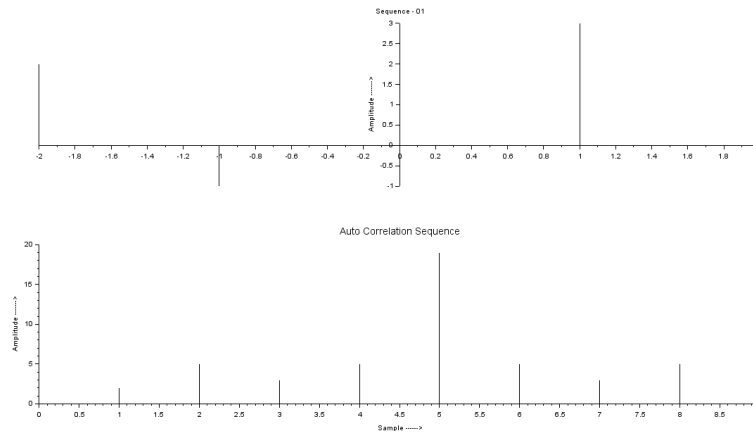


Figure 9.1: Auto correlation

```

// -2
14 e1=s1+n-1;
15 n1=s1:1:e1;
16 subplot (2 ,1 ,1)
17 plot2d3 ('gmn',n1,x1);
18 ylabel ("Amplitude ----->");
19 title ("Sequence - 01 ");
20 a = gca ();
21 a.x_location = "origin";
22 a.y_location = "origin";
23 x2 = x1($:-1:1);
24 n1 = s1+s1;
25 nh = n1+n1+n1-2;
26 c= convol(x1 ,x2) // function for Auto correlation
27 disp(c,'Cross Correlation Sequence is =')
28 subplot (2 ,1 ,2)
29 plot2d3 ('gmn',c)
30 a = gca ();
31 a.x_location = "origin";
32 a.y_location = "origin";
33 xlabel ("Sample ----->");
34 ylabel ("Amplitude ----->");
35 title ("Auto Correlation Sequence" ,"fontsize" ,3);

```

```
36
37 //..... Execution in Consol .....
38
39 //ENTER THE FIRST SEQUENCE 01 [use square bracket-
    array form]= [2 -1 2 3 1]
40
41 //ENTER THE starting point of array = -2
42
43 // Cross Correlation Sequence is =
44
45 //  2.   5.   3.   5.   19.   5.   3.   5.   2.
```

---

## Experiment: 10

Develop a program to obtain Z transform of basic function using Scilab.

Scilab code Solution 10.01 Z transform

```
1 //Experiment-10
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM:  Develop a program To obtain Z transform using
      SCILAB code
7
8
9 clear ;
10 clc ;
11 x= input ("Negative sequence coefficient =")//[1 2 3
      2 7]
12 T= input("sampling time =") // 1
13 Xz= poly([x],"Z","coeff")
14 disp (Xz, 'X(Z) =')
15 //For Positive sequence
```

```

16 z= poly (0, 'z')
17 disp ("positive sequence of Z transform =")
18 Xzp = horner (Xz ,1/z)
19 disp (Xzp, 'X(z)' )
20
21 // Discreat form
22 disp("Transfer Function in State-space
      representation = ")
23 s= poly(0, 's');
24 z= poly(0, 'z');
25 tf= syslin('c',(s+1)/(s^2-5*s+2));
26 disp (tf)
27 disp ("Transfer Function in Discrete form = ")
28 df = horner(tf,(2/T)*(z -1)/(z+1))
29 disp (df)
30
31
32 // .... Execution Console ....
33
34 //Negative equence coefficient =[1,2,3,2,7]
35 //sampling time =1
36
37 // X(Z) =
38
39 //           2      3      4
40 //   1 + 2Z + 3Z + 2Z + 7Z
41
42 // positive sequence coefficient =
43
44 //X(z)
45
46 //           2      3      4
47 //   7 + 2z + 3z + 2z + z
48 //   -----
49 //           4
50 //           z
51
52 //Transfer Function in State-space representation

```

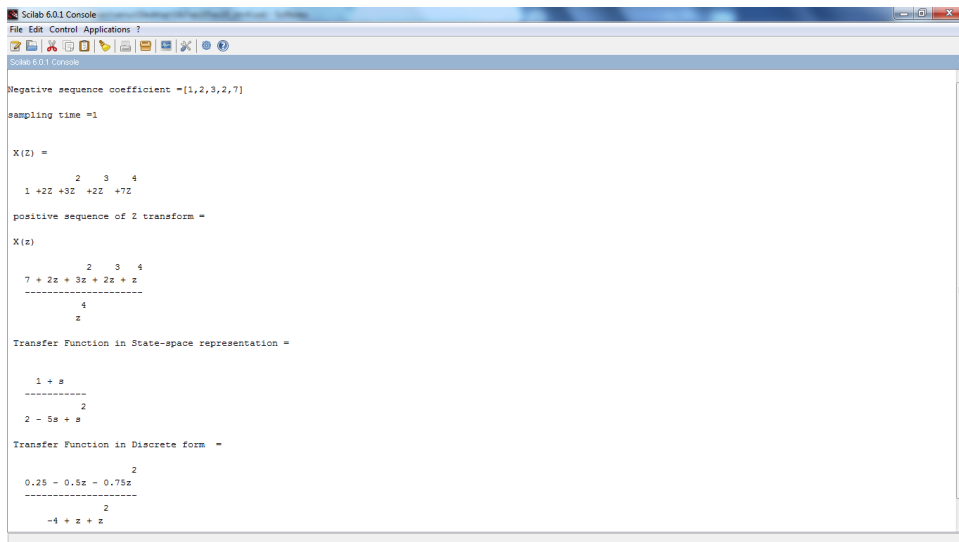


Figure 10.1: Z transform

```

53 // =
54
55 // 1 + s
56 // -----
57 //                               2
58 // 2 - 5s + s
59
60 // Transfer Function in Discrete form =
61
62 //                               2
63 // 0.25 - 0.5z - 0.75z
64 // -----
65 //                               2
66 // - 4 + z + z

```

---

## Experiment: 11

Develop a program to obtain pole zero plot of given transfer function using Scilab.

Scilab code Solution 11.01 Pole Zero form Transfer function

```
1 //Experiment-11
2 // windows - 7 - 64-Bit
3 //Scilab - 5.4.1
4
5
6 //AIM: Develop a program to Plot pole zero fro
   transfer function using SCILAB code
7
8
9 clear ;
10 clc ;
11 z=poly(0, 'z');
12 num=input('Enter Numerator equation =')//[1];
13 den=input('Enter Denominator equation =')//[1-1.5*z
   ^-1+0.5*z^-2];
14 tf=num./den
15 disp(tf,"Transfer Function with positive power = ")
```



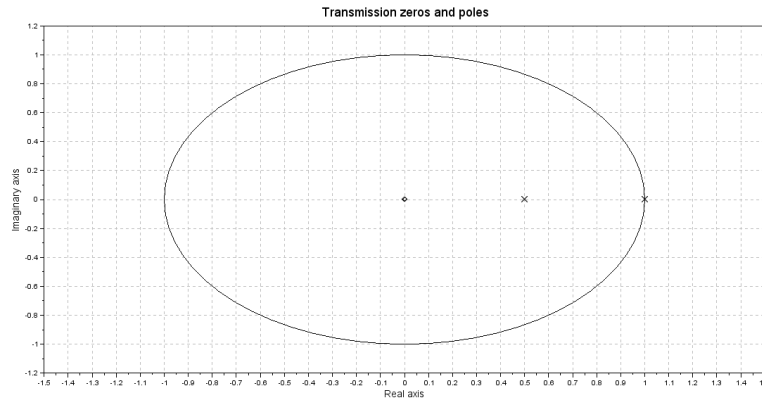


Figure 11.1: Pole Zero form Transfer function

```

16 H=syslin('c',tf);
17 plzr(H)
18 scf;
19 bode(H)
20
21 //——— COnsole window———
22 //Enter Numerator equation =[1]
23 //Enter Denominator equation =[1-1.5*z^-1+0.5*z^-2]
24
25 //Transfer Function with positive power =
26
27 //          2
28 //          z
29 //  —————
30 //                                2
31 //  0.5 - 1.5z + z

```

---

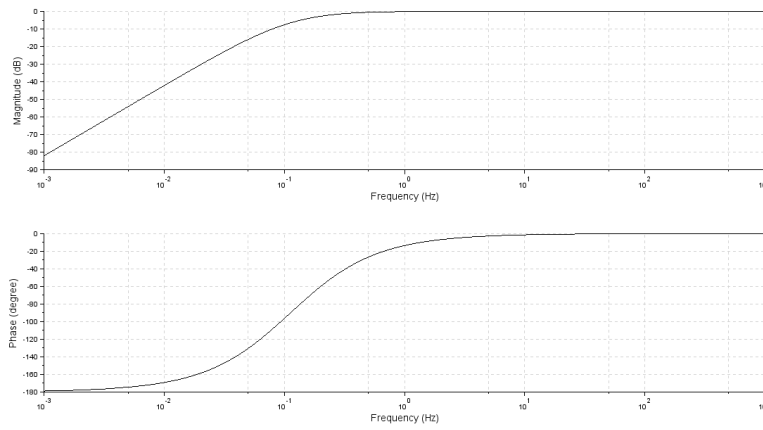


Figure 11.2: Pole Zero form Transfer function

## Experiment: 12

Develop a program to understand the concept of amplitude modulation.

Scilab code Solution 12.01 Amplitude Modulation

```
1 //Experiment -12
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM:  Develop a program for Performing Amplitude
      modulation using SCILAB code
7
8
9 clear ;
10 clc;
11 f= input ("Enter the value of signal frequency=");
      // f=5
12 fc=input (" Enter the value of carrier frequency=");
      //(fc>f) fc = 100
```

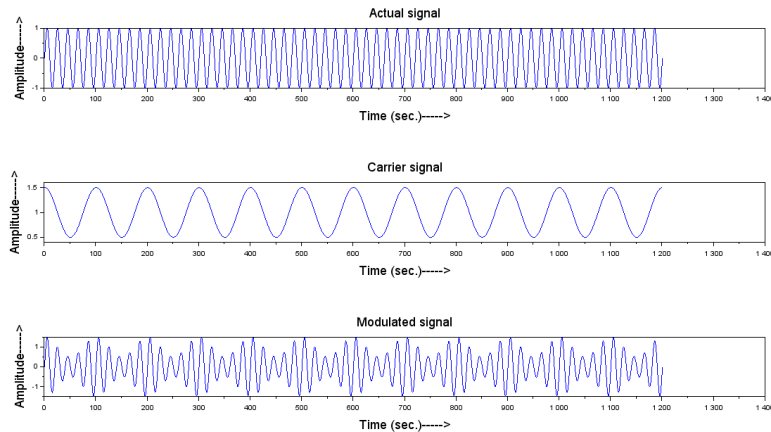


Figure 12.1: Amplitude Modulation

```

13 n=0:1200;
14 t=n/fc;
15 m=input("Enter the value of modulation index ="); //
    m = 1
16 s1=sin(2*%pi*f*t);
17 subplot(311);
18 plot(s1);
19 xlabel("Time (sec.)——>" ," fontsize",4);
20 ylabel("Amplitude——>" ," fontsize",4);
21 title("Actual signal" ," fontsize",4);
22 //Generation of carrier signal
23 s2=1+0.5*cos(2*%pi*m*t);
24 subplot (312);
25 plot (s2);
26 xlabel("Time (sec.)——>" ," fontsize",4);
27 ylabel("Amplitude——>" ," fontsize",4);
28 title("Carrier signal" ," fontsize",4);
29 ms=s2.*s1; // Modulating Signal
30 subplot (313);
31 plot(ms);
32 xlabel("Time (sec.)——>" ," fontsize",4);
33 ylabel("Amplitude——>" ," fontsize",4);

```

```
34 title("Modulated signal" ,"fontsize",4);
35
36 //—— consol value——
37 //Enter the value of signal frequency=5
38
39 //Enter the value of carrier frequency=100
40
41 //Enter the value of modulation index =1
```

---

## Experiment: 13

Develop a program to understand the concept of frequency modulation.

Scilab code Solution 13.01 Frequency Modulation

```
1 //Experiment-13
2 // windows - 7 - 64-Bit
3 //Scilab - 6.0.1
4
5
6 //AIM:  Develop a program for Performing Frequency
      modulation using SCILAB code
7
8
9 clear ;
10 clc;
11 t=0:0.0001:0.1;
12 vm=input ("Enter the value of modulating volatge =")
      ); // vm = 5
13 fm=input ("Enter the value of modulating frequency="
```

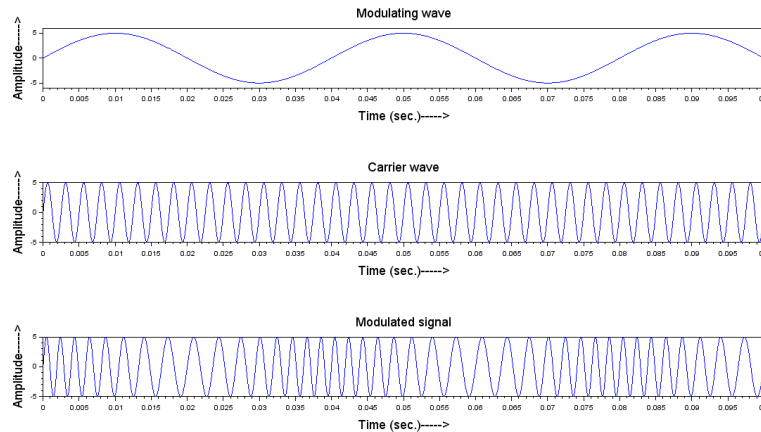


Figure 13.1: Frequency Modulation

```

); // fm =25
14 vc=input ("Enter the value of carrier volatge =");
    // vc = 5
15 fc=input (" Enter the value of carrier frequency=");
    //(fc>f) fc = 400
16 m=input("Enter the value of modulation index ="); //
    m = 5
17 //Generation of carrier signal
18 v1=vc*sin(2*%pi*fc*t);
19 subplot (312);
20 plot (t,v1);
21 xlabel("Time (sec.)——>" ," fontsize",4);
22 ylabel(" Amplitude——>" ," fontsize",4);
23 title(" Carrier wave" ," fontsize",4);
24 //generation of Modulating wave
25 v2=vm*sin(2*%pi*fm*t);
26 subplot(311);
27 plot (t,v2);
28 xlabel("Time (sec.)——>" ," fontsize",4);
29 ylabel(" Amplitude——>" ," fontsize",4);
30 title(" Modulating wave " ," fontsize",4);
31 vfm = 5*(sin(2*%pi*fc*t+(m.*sin(2*%pi*fm*t)))) //

```

```

    Modulating Signal
32 subplot (313);
33 plot(t,vfm);
34 xlabel("Time (sec.)——>" ,"fontsize",4);
35 ylabel("Amplitude——>" ,"fontsize",4);
36 title("Modulated signal" ,"fontsize",4);
37
38 //—— consol value——
39
40 //Enter the value of modulating volatge =5
41
42 //Enter the value of modulating frequency=25
43
44 //Enter the value of carrier volatge =5
45
46 // Enter the value of carrier frequency=400
47
48 //Enter the value of modulation index =5

```

---