Scilab Manual for
Signals and Systems
by Dr Dr. R. Anjana
Electrical Engineering
Laxmi Institute Of Technology, Sarigam, Gujarat[1]

Solutions provided by
Dr Dr. R. Anjana
Electrical Engineering
Laxmi Institute Of Technology, Sarigam, Gujarat

May 14, 2026

Contents

List of Experiments

List of Figures

Experiment: 1

Waveform generation for continuous signals

Scilab code Solution 1.01 PROGRAM TO GENERATE COMMON CONTINUOUS TIME SIGNALS

```
1  // VERSION: Scilab: 5.4.1
2  // OS: windows 7
3  // CAPTION: PROGRAM TO GENERATE COMMON CONTINUOUS
     TIME SIGNALS
4
5  //UNIT IMPULSE SIGNAL
6  clc;
7  clear all;
8  close;
9  N=5; //SET LIMIT
10 t1=-5:5;
11 x1=[zeros(1,N),ones(1,1),zeros(1,N)];
12 subplot(2,3,1);
13 plot(t1,x1)
14 xlabel('time');
15 ylabel('Amplitude');
```

Figure 1.1: PROGRAM TO GENERATE COMMON CONTINUOUS TIME SIGNALS

```scilab
16  title ('Unit impulse signal ');
17
18
19
20  //UNIT STEP SIGNAL
21  t2 =0:4;
22  x2=ones (1 ,5);
23  subplot (2 ,3 ,2);
24  plot (t2 ,x2);
25  xlabel ('time ');
26  ylabel ('amplitude ');
27  title ('Unit Step Continuous Signal ');
28
29  title ('Unit step signal ');
30
31  //EXPONENTIAL SIGNAL
32  t3 =0:1:20;
33  x3=exp (-t3);
34  subplot (2 ,3 ,3);
35  plot (t3 ,x3);
36  xlabel ('time ');
37  ylabel ('Amplitude ');
38  title ('Exponential signal ');
39
40
41
42  //UNIT RAMP SIGNAL
43  t4 = -0:20;
44  x4=t4;
45  subplot (2 ,3 ,4);
46  plot (t4 ,x4);
47  xlabel ('time ');
48  ylabel ('Amplitude ');
49  title ('Unit ramp signal ');
50
51  //SINUSOIDAL SIGNAL
52  t5 =0:0.04:1;
53  x5=sin (2* %pi*t5);
```

```
54  subplot(2,3,5);
55  plot(t5,x5);
56  title('Sinusoidal Signal')
57  xlabel('time');
58  ylabel('Amplitude');
59
60  //RANDOM SIGNAL
61  t6=-10:1:20;
62  x6=rand(1,31);
63  subplot(2,3,6);
64  plot(t6,x6);
65  xlabel('time');
66  ylabel('Amplitude');
67  title('Random signal');
```

Experiment: 2

Waveform generation for Discrete signals

Scilab code Solution 2.02 PROGRAM TO GENERATE COMMON DISCRETE TIME SIGNALS

```
1  // VERSION: Scilab: 5.4.1
2  // OS: windows 7
3  // CAPTION: PROGRAM TO GENERATE COMMON DISCRETE TIME
       SIGNALS
4
5  //UNIT IMPULSE SIGNAL
6  clc;
7  clear all;
8  close;
9  N=5; //SET LIMIT
10 t1=-5:5;
11 x1=[zeros(1,N),ones(1,1),zeros(1,N)];
12 subplot(2,4,1);
13 plot2d3(t1,x1)
14 xlabel('time');
15 ylabel('Amplitude');
```
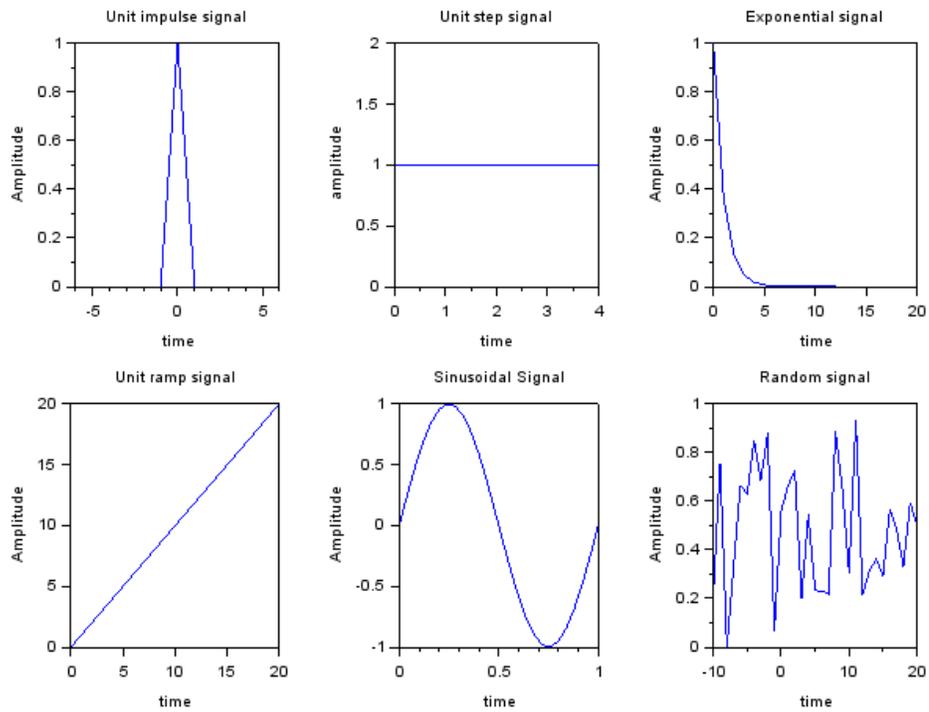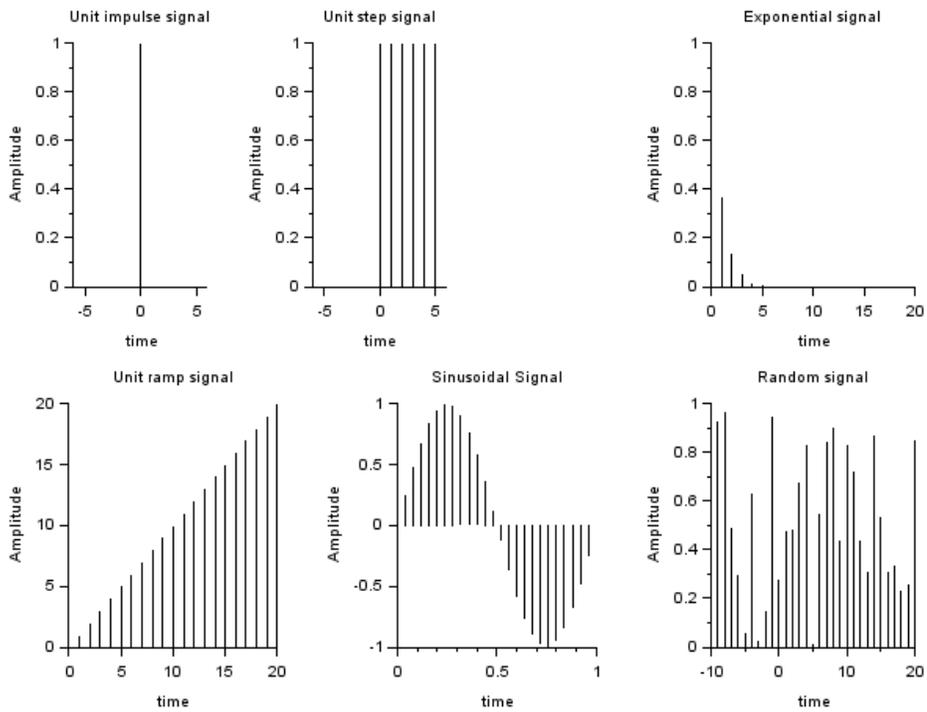
Figure 2.1: PROGRAM TO GENERATE COMMON DISCRETE TIME SIGNALS

```
16  title('Unit impulse signal');
17
18
19  //UNIT STEP SIGNAL
20  t2=-5:5;
21  x2=[zeros(1,N),ones(1,N+1)];
22  subplot(2,4,2);
23  plot2d3(t2,x2)
24  xlabel('time');
25  ylabel('Amplitude');
26  title('Unit step signal');
27
28  //EXPONENTIAL SIGNAL
29  t3=0:1:20;
30  x3=exp(-t3);
31  subplot(2,3,3);
32  plot2d3(t3,x3);
33  xlabel('time');
34  ylabel('Amplitude');
35  title('Exponential signal');
36
37
38
39  //UNIT RAMP SIGNAL
40  t4=0:20;
41  x4=t4;
42  subplot(2,3,4);
43  plot2d3(t4,x4);
44  xlabel('time');
45  ylabel('Amplitude');
46  title('Unit ramp signal');
47
48  //SINUSOIDAL SIGNAL
49  t5=0:0.04:1;
50  x5=sin(2*%pi*t5);
51  subplot(2,3,5);
52  plot2d3(t5,x5);
53  title('Sinusoidal Signal')
```

```
54 xlabel('time');
55 ylabel('Amplitude');
56
57 //RANDOM SIGNAL
58 t6=-10:1:20;
59 x6=rand(1,31);
60 subplot(2,3,6);
61 plot2d3(t6,x6);
62 xlabel('time');
63 ylabel('Amplitude');
64 title('Random signal');
```

Experiment: 3

Basic Operations on DT signals

Scilab code Solution 3.03 TO PERFORM BASIC OPERATIONS ON DT SIGNALS

```
1  // VERSION: Scilab: 5.4.1
2  // OS: windows 7
3
4  //CAPTION:TO PERFORM BASIC OPERATIONS ON D.T SIGNALS
5
6  clc;
7  clear all;
8  close;
9
10 //amplification
11 x=input('Enter input sequence x:');
12 a=input('Enter amplification factor a:');
13 b=input('Enter attenuation factor b:');
14 c=input('Enter amplitude reversal factor c:');
15 y1=a*x;
16 y2=b*x;
17 y3=c*x;
18 n=length(x);
```
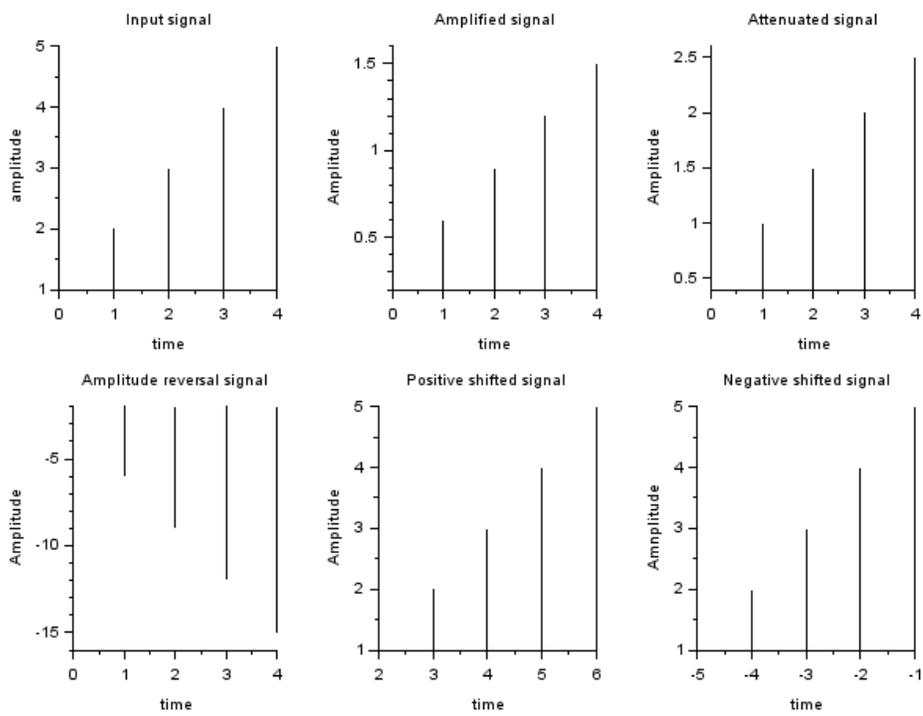
Figure 3.1: TO PERFORM BASIC OPERATIONS ON DT SIGNALS

```
19
20  //Input signal plot
21  subplot(2,3,1);
22  plot2d3(0:n-1,x);
23  xlabel('time');
24  ylabel('amplitude');
25  title('Input signal');
26
27  //Amplification
28  subplot(2,3,2);
29  plot2d3(0:n-1,y1);
30  xlabel('time');
31  ylabel('Amplitude');
32  title('Amplified signal');
33
34  //attenuation
35  subplot(2,3,3);
36  plot2d3(0:n-1,y2);
37  xlabel('time');
38  ylabel('Amplitude');
39  title('Attenuated signal');
40
41  //Amplitude Reverasl
42  subplot(2,3,4);
43  plot2d3(0:n-1,y3);
44  xlabel('time');
45  ylabel('Amplitude');
46  title('Amplitude reversal signal');
47
48  // folding and Shifting
49
50  n0=input('Enter the +ve shift:');
51  n1=input('Enter the -ve shift:');
52  l=length(x);
53  i=n0:l+n0-1;
54  j=n1:l+n1-1;
55  subplot(2,3,5);
56  plot2d3(i,x);
```

```
57  xlabel('time');
58  ylabel('Amplitude');
59  title('Positive  shifted  signal');
60  subplot(2,3,6);
61  plot2d3(j,x);
62  xlabel('time');
63  ylabel('Amnplitude');
64  title('Negative  shifted  signal');
65
66
67  //**************************//
68  //INPUT:  In  Console  Window
69  //  **********************//
70
71  //Enter  the  Input  Sequence  x(n)=[1  2  3  4  5]
72  //Enter  the  amplification  factor  a = 0.3
73  //Enter  the  attenuation  factorb = 0.5
74  //Enter  the  amplitude  reversal  factor  c = -3
75  //Enter  the  positive  shift  : 2
76  //Enter  the  negative  shift  : -5
77
78   //OUTPUT:  In  Graphic  Windows
```

Experiment: 4

ADDITION AND MULTIPLICATION ON DT SIGNALS

Scilab code Solution 4.04 To perform addition and subtraction of the following two DT signals

```
1  // VERSION: Scilab: 5.4.1
2  // OS: windows 7
3
4  //CAPTION: To perform addition and subtraction of
       the following two DT signals
5  //Xa(t) =     1; 0<t<1
6  //            2; 1<t<2;
7  //            1; 2<t<3
8
9  // &
10 // Xb(t) = t; 0<t<1
11 //         1; 1<t<2;
12 //         3-t; 2<t<3
13
```
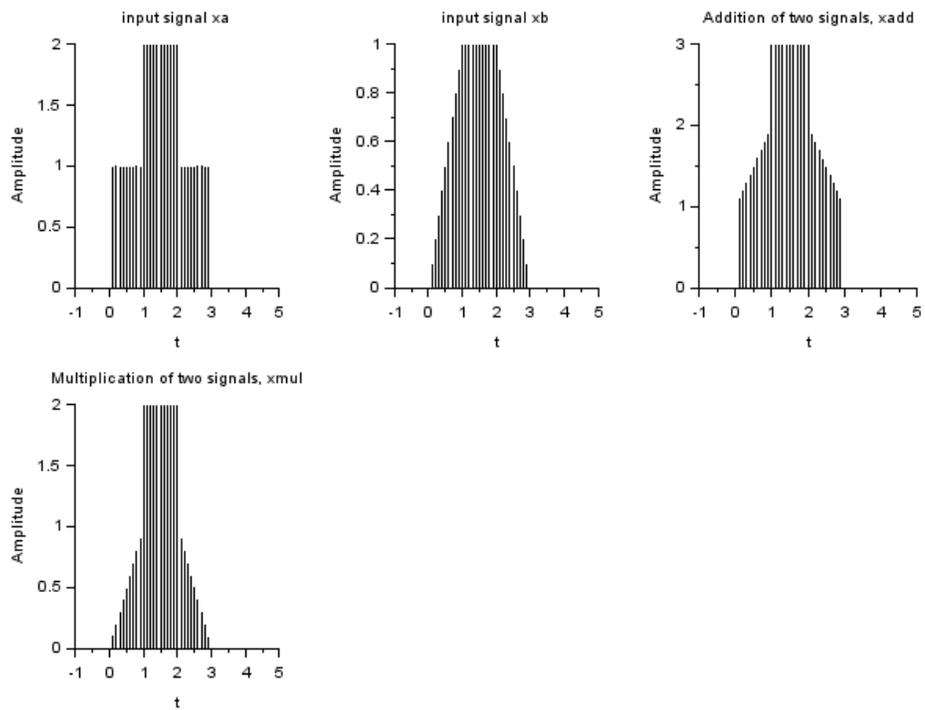
Figure 4.1: To perform addition and subtraction of the following two DT signals

```scilab
14  //
    ////*******************************////////////////////////////

15
16  clc;
17  clear all;
18  close
19  t = -1:0.1:5;
20  x1 = 1;
21  x2 = 2;
22  x3 = 3-t;
23
24  xa = x1.*(t>0 & t<1) + x2.*(t>=1 & t<=2) + x1.*(t>2
     & t<3);
25  xb = t.*(t>0 &t<1) + x1*(t>=1 &t<=2)+x3.*(t>2 &t<3);
26
27  xadd = xa +xb;
28  xmul = xa.*xb;
29  subplot(2,3,1);
30  plot2d3(t,xa)   //Plots input signal xa
31  xlabel('t');
32  ylabel('Amplitude');
33  title('input signal xa');
34  subplot(2,3,2);
35  plot2d3(t,xb)   //Plots input signal xb
36  xlabel('t');
37  ylabel('Amplitude');
38  title('input signal xb');
39  subplot(2,3,3);
40  plot2d3(t,xadd)   //Plots addition of signal xa and
      xb
41  xlabel('t');
42  ylabel('Amplitude');
43  title('Addition of two signals, xadd');
44  subplot(2,3,4);
45  plot2d3(t,xmul)   //Plots Multiplication of signal xa
       and xb
46  xlabel('t');
```

```
47  ylabel('Amplitude');
48  title('Multiplication of two signals, xmul');
49
50  //// Output : In Graphic Window
```

Experiment: 5

EVEN and ODD SIGNALS

Scilab code Solution 5.05 EVEN and ODD SIGNALS

```
1  // VERSION: Scilab: 5.4.1
2  // OS: windows 7
3
4  //CAPTION: EVEN and ODD SIGNALS for x(t)=sint(t)+cos
     (t)
5
6  clc;
7  close;
8  clear all;
9  t=0:.005:4*%pi;
10 x=sin(t)+cos(t); // Given signal: x(t)=sint(t)+cos(t
     )
11 subplot(2,2,1)
12 plot(t,x)
13 xlabel('t');
14 ylabel('amplitude')
15 title('input signal')
16 y=sin(-t)+cos(-t) // Put t = -t in x(t)
17 subplot(2,2,2)
```
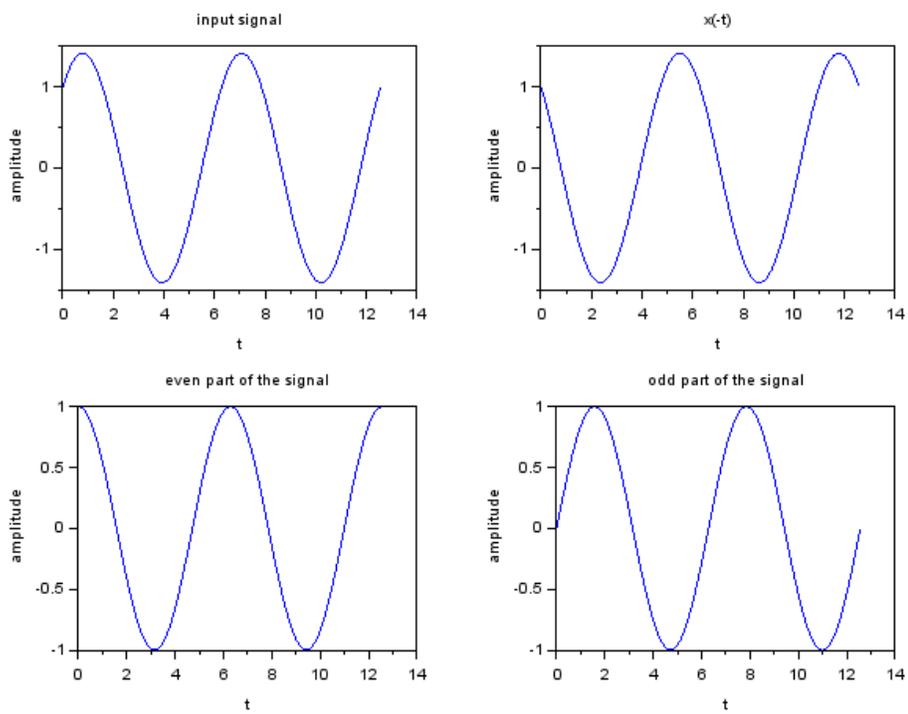
Figure 5.1: EVEN and ODD SIGNALS

```
18  plot(t,y)
19  xlabel('t');
20  ylabel('amplitude')
21  title('x(-t)')
22  z=x+y
23  subplot(2,2,3)
24  plot(t,z/2)   // to plot even signal
25  xlabel('t');
26  ylabel('amplitude')
27  title('even part of the signal')
28  p=x-y
29  subplot(2,2,4)
30  plot(t,p/2) // to plot odds signal
31  xlabel('t');
32  ylabel('amplitude');
33  title('odd part of the signal');
34
35  //Output: In graphic window
```

Experiment: 6

Linear and Non-linear Signals

Scilab code Solution 6.06 To find wheather the system is linear or non linear for the given signal

```
1  // VERSION: Scilab: 5.4.1
2  // OS: windows 7
3
4  //CAPTION: To find wheather the system is linear or
       non linear for the given signal y(n)=[x(n)]^2+B;
5
6  clc;
7  clear all;
8  close;
9
10  //Properties of DT Systems(Linearity)
11  //y(n)=[x(n)]^2+B;
12
13  x1=input('Enter first input sequence:');
14  n=length(x1);
15  x2=input('Enter second input sequence:');
16  a=input('Enter scaling constant(a):');
17  b=input('Enter scaling constant(b):');
```
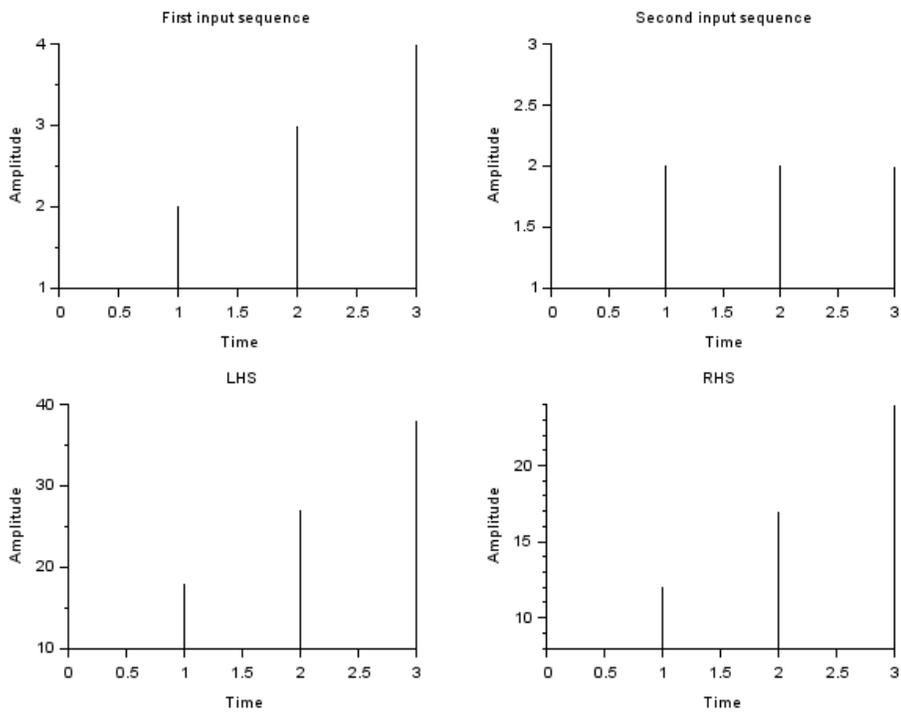
Figure 6.1: To find wheather the system is linear or non linear for the given signal

26

```
18  B=input('Enter scaling constant(B):');
19
20  y1=x1.^2+B;
21  y2=x2.^2+B;
22  rhs=a*y1+b*y2;
23  x3=a*x1+b*x2;
24  lhs=x3.^2+B;
25
26  subplot(2,2,1);
27  plot2d3(0:n-1,x1);
28  xlabel('Time');
29  ylabel('Amplitude');
30  title('First input sequence');
31  subplot(2,2,2);
32  plot2d3(0:n-1,x2);
33  xlabel('Time');
34  ylabel('Amplitude');
35  title('Second input sequence');
36  subplot(2,2,3);
37  plot2d3(0:n-1,lhs);
38  xlabel('Time');
39  ylabel('Amplitude');
40  title('LHS');
41  subplot(2,2,4);
42  plot2d3(0:n-1,rhs);
43  xlabel('Time');
44  ylabel('Amplitude');
45  title('RHS');
46
47  if(lhs==rhs)
48      disp('system is linear');
49  else
50      disp('system is non-linear');
51
52  end;
53
54
55  //// Input Data
```

```
56
57  //Enter first input sequence:[1 2 3 4]
58  //Enter second input sequence:[ 2 2 2 2 ]
59  //Enter scaling constant(a):1
60  //Enter scaling constant(b):1
61  //Enter scaling constant(B):2
62
63  /// Output:  system is non-linear
64  //See graphic window
```

Experiment: 7

Energy and Power Signal

Scilab code Solution 7.7 To find the energy and power of a signal

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3
4 //CAPTION: To find the energy and power of a signal
5
6
7 //To find energy of a signal
8 clc;
9 clear all;
10 close;
11  E =0;
12  for n =0:50
13  x ( n +1) =( 0.5) ^ n ;
14  end
15  for n =0:100
16  E = E + x (n +1) ^2;
17  end
18  if E < %inf then
19  disp (E ,'The Energy of the given signal is E = ')
       ;
20  else
```

```
21  disp ( 'The given signal is energy signal is not an
        Energy Signal ' ) ;
22  end
23
24  ///// To find power of the Signal
25  T=10;        //Total evaluation time
26  Ts=0.001;    //Sampling time =&gt; 1000 samples per
        second
27  Fs=1/Ts;     //Sampling period
28  t=[0:Ts:T]; //define simulation time
29  x=cos(2*%pi*100*t)+cos(2*%pi*200*t)+sin(2*%pi*300*t)
        ;
30  power = (norm(x)^2)/length(x);
31  disp(power, 'The power of the given signal is P = ')
32
33
34  // output
35  //The Energy of the given signal is E =51.333333
36
37  // The power of the given signal is P = 1.50025
```

Experiment: 8

Time Variance and Time Invariance

Scilab code Solution 8.08 To find wheather the signal is time variant or time invariant for the signal

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3
4 //CAPTION: To find wheather the signal is time
       variant or time invariant for the signal y(n)=n*[
       x(n)];
5 clc;
6 clear all;
7 close ;
8  x1=input('Enter input sequence x1:');
9 n1=length(x1);
10 for n=1:n1
11     y1(n1)=n.*x1(n);
12 end;
13 n0=input('Enter shift:');
14 x2=[zeros(1,n0),x1];
```
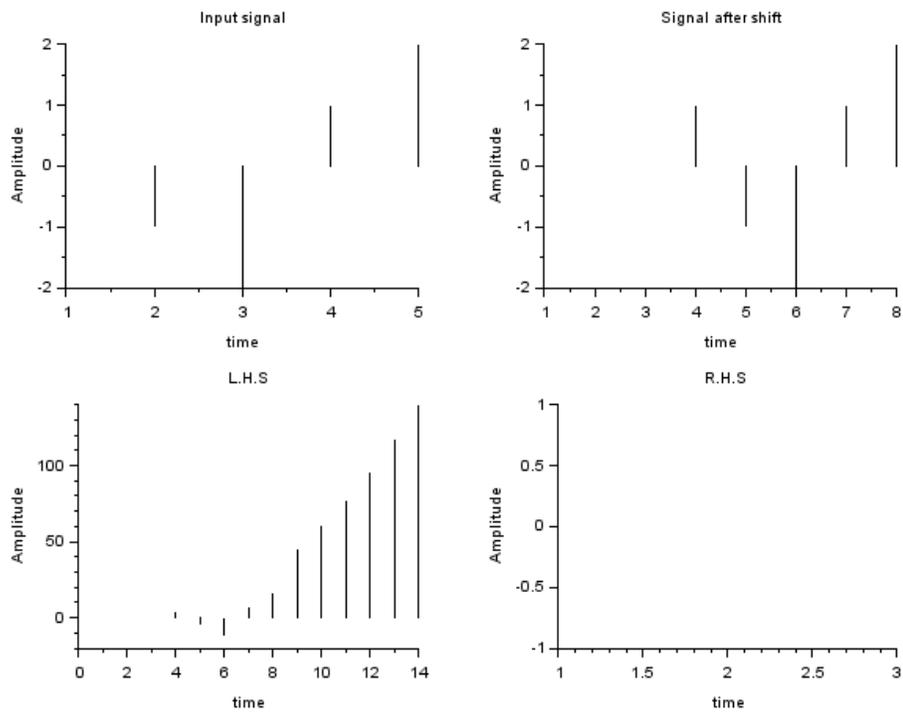
Figure 8.1: To find wheather the signal is time variant or time invariant for the signal

```
15  for n2 =1: n1+n0
16      y2 ( n2 )=n2 .* x2 ( n2 );
17  end ;
18  y3 =[ zeros (1 , n0 )] , y1 ;
19   if ( y2 == y3 )
20      disp ( 'system  is  time  invariant ');
21  else
22      disp ( 'system  is  time  variant ');
23  end ;
24  subplot (2 ,2 ,1);
25  plot2d3 ( x1 );
26  xlabel ( 'time ');
27  ylabel ( 'Amplitude ');
28  title ( 'Input  signal ');
29  subplot (2 ,2 ,2);
30  plot2d3 ( x2 );
31  xlabel ( 'time ');
32  ylabel ( 'Amplitude ');
33  title ( 'Signal  after  shift ');
34  subplot (2 ,2 ,3);
35  plot2d3 ( y2 );
36  xlabel ( 'time ');
37  ylabel ( 'Amplitude ');
38  title ( 'L.H.S ');
39  subplot (2 ,2 ,4);
40  plot2d3 ( y3 );
41  xlabel ( 'time ');
42  ylabel ( 'Amplitude ');
43  title ( 'R.H.S ');
44
45
46  //// Sample Input
47  ///Enter input sequence x1 :[1 -1 -2 1 2]
48  //Enter shift :3
49
50  // Output
51  //system is time variant
```

Experiment: 9

DISCRETE CONVOLUTION

Scilab code Solution 9.09 To perform discrete convolution for two given sequences

```
1  // VERSION:  Scilab :  5.4.1
2  // OS:  windows  7
3
4  //CAPTION:  To  perform  discrete  convolution  for  two
       given  sequences
5
6  clc ;
7  clear  all ;
8  close  ;
9  a=input ( 'Enter  the  starting  point  of  x [ n]= ') ;
10   b=input ( 'Enter  the  starting  point  of  h [ n]= ') ;
11  x=input ( 'Enter  the  co-efficients  of  x [ n]= ') ;
12  h=input ( 'Enter  the  co-efficients  of  h [ n]= ') ;
13  y=conv ( x , h ) ;
14  subplot (3 ,1 ,1) ;
15   p=a : ( a+length ( x ) -1) ;
16  plot2d3 ( p , x ) ;
17   xlabel ( 'Time ') ;
```
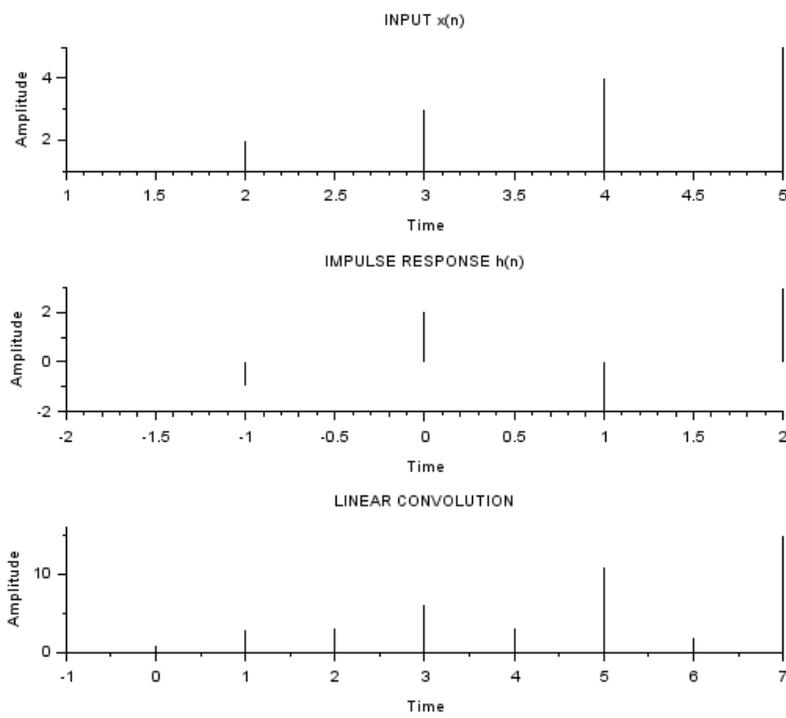
Figure 9.1: To perform discrete convolution for two given sequences

```
18    ylabel('Amplitude');
19    title('INPUT x(n)');
20    subplot(3,1,2);
21      q=b:(b+length(h)-1);
22  plot2d3(q,h);
23    xlabel('Time');
24  ylabel('Amplitude');
25  title('IMPULSE RESPONSE h(n)');
26  subplot(3,1,3);
27   n=a+b:length(y)+a+b-1;
28    plot2d3(n,y);
29
30  disp(y)
31  xlabel('Time');
32  ylabel('Amplitude');
33  title('LINEAR CONVOLUTION');
34
35  ///Sample Input
36  //Enter the starting point of x[n]=1
37  //Enter the starting point of h[n]=-2
38  //Enter the co-efficients of x[n]=[1 2 3 4 5]
39  //Enter the co-efficients of h[n]=[1 -1 2 -2 3]
40
41  // output: See the Graphic window
```

Experiment: 10

Z- TRANSFORMS AND INVERSE Z - TRANFORM

Scilab code Solution 10.10 To find Z Transform and inverse Z Transform for the given sequence

```
1
2 // VERSION: Scilab: 5.4.1
3 // OS: windows 7
4
5 //CAPTION: To find Z-Transform and inverse Z-
     Transform for the given sequence
6
7 // Z transform of given sequence
8 clear all;
9 clc ;
10  close ;
11  z = poly (0 , 'z' , 'r');
12  x1 = input('enter the input sequence:');
13  n1 = 0: length ( x1 ) -1;
14 X1 = x1.*[(1/ z ) .^n1];
15 disp(X1,'the z-transform of X1: ')
16
17
```

```
18  // Inverse Z-transform
19
20
21   z=%z;
22   a =(2+2* z+z ^2) ;
23  b=z^2;
24   h = ldiv (b,a ,6);
25  disp (h," The Inverse Z - Transform is");
26
27
28
29  // Sample input data
30
31  //enter the input sequence:[1 2 3 4 5]
32
33
34
35  // Output Data
36
37  //the z-transform of X1:
38
39   //    1      2      3      4      5
40   //    -      -      -      -      -
41  ////                   2      3      4
42   //    1      z      z      z      z
43
44   // The Inverse Z - Transform is
45
46   //    1.
47   // - 2.
48   //    2.
49   //    0.
50  //  - 4.
51   //    8.
```

Experiment: 11

Fourier Transform

Scilab code Solution 11.11 Fourier Transform

```
1  // VERSION: Scilab: 5.4.1
2  // OS: windows 7
3
4  //CAPTION:A symmetrical rectangular pulse is given
       by R(n) = 1 ; -N<n<N = 0 ; otherwise;
5  //Determine The DTFT for N = {5,15,25,100}. Scale
       the DTFT so that x(0) = 1 ,also plot the
       normalized DTFT over -   to   .
6
7
8  clc;
9  clear all;
10 close
11 N=input('Enter The Value Of N1 = ');
12 n=-N:1:N;
13 x=ones(1,length(n));
14 w=(-%pi:(%pi/100):%pi);
15 z=x*exp(-%i*n'*w);
16 subplot(2,2,1);
```
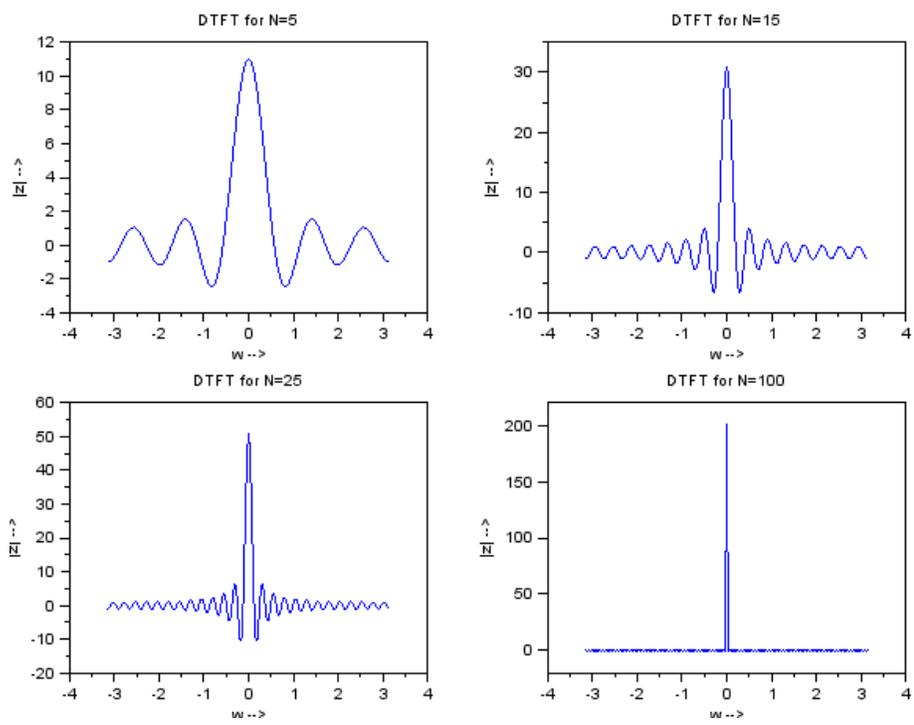
Figure 11.1: Fourier Transform

```
17  plot(w,z);
18  xlabel('w --->');
19  ylabel('|z| --->');
20  title('DTFT for N=5');
21
22  N=input('Enter The Value Of N2 = ');
23  n=-N:1:N;
24  x=ones(1,length(n));
25  w=(-%pi:(%pi/100):%pi);
26  z=x*exp(-%i*n'*w);
27  subplot(2,2,2);
28  plot(w,z);
29  xlabel('w --->');
30  ylabel('|z| --->');
31  title('DTFT for N=15');
32
33  N=input('Enter The Value Of N3 = ');
34  n=-N:1:N;
35  x=ones(1,length(n));
36  w=(-%pi:(%pi/100):%pi);
37  z=x*exp(-%i*n'*w);
38  subplot(2,2,3);
39  plot(w,z);
40  xlabel('w --->');
41  ylabel('|z| --->');
42  title('DTFT for N=25');
43
44  N=input('Enter The Value Of N4 = ');
45  n=-N:1:N;
46  x=ones(1,length(n));
47  w=(-%pi:(%pi/100):%pi);
48  z=x*exp(-%i*n'*w);
49  subplot(2,2,4);
50  plot(w,z);
51  xlabel('w --->');
52  ylabel('|z| --->');
53  title('DTFT for N=100');
54
```

```
55  ////////////////////////
56  /// SAMPLE INPUTS//////
57
58  //Enter The Value Of N1 = 5
59  //Enter The Value Of N2 = 15
60  //Enter The Value Of N3 = 25
61  //Enter The Value Of N4 = 100
```

Experiment: 12

Fourier Series

Scilab code Solution 12.12 Write a program to find the fourier series coefficients of a signal

```
1
2  // VERSION: Scilab: 5.4.1
3  // OS: windows 7
4
5  //CAPTION: Write a program to find the fourier
       series coefficients of a signal.
6
7  fig_size = [232 84 774 624];
8  x = [0.1 0.9 0.1];// % 1 period of x(t)
9  x = [x x x x]; //% 4 periods of x(t)
10 tx = [-2 -1 0 0 1 2 2 3 4 4 5 6]; //% time points
       for x(t)
11 subplot(2,2,1)
12 plot(tx,x)
13 xlabel('Time (s)')
14 ylabel('Amplitude'),...
15 title('Periodic Signal x(t)')
16
```
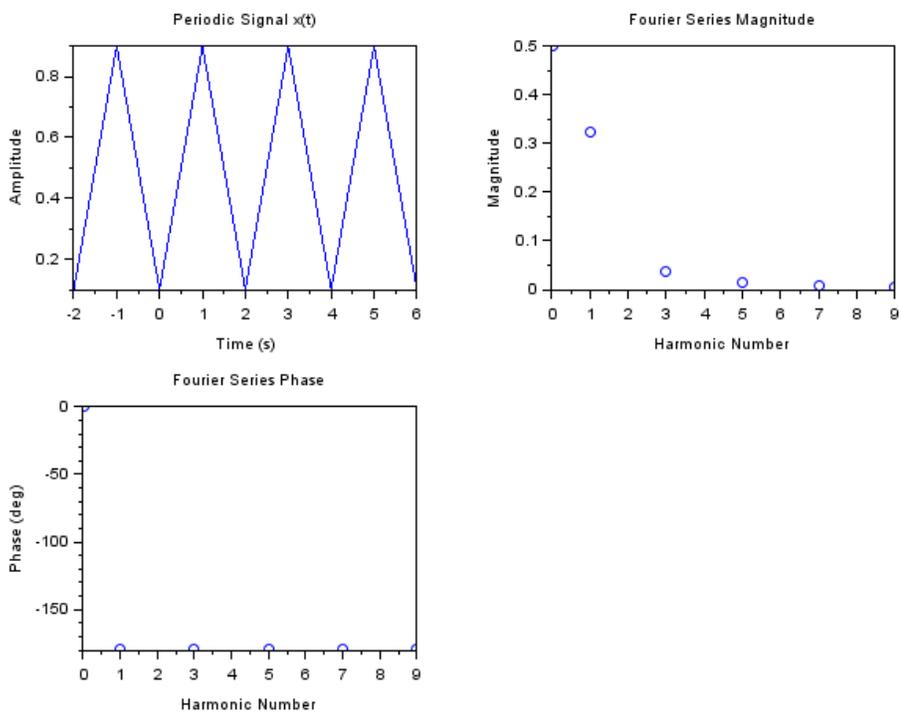
Figure 12.1: Write a program to find the fourier series coefficients of a signal

```
17  a0 = 0.5; // DC component of Fourier Series
18  ph0 = 0;
19  n = [1 3 5 7 9]; // Values of n to be evaluated
20  an = -3.2 ./ (%pi * n).^2; //% Fourier Series
        coefficients
21  mag_an = abs(an);
22  ph_an = -180 * ones(1,length(n));
23  n = [0 n];
24  mag_an = [a0 mag_an]; // % Including a0 with a_n
25  ph_an = [ph0 ph_an];
26
27  //Plotting fourier series magnitude
28  subplot(2,2,2)
29  plot(n,mag_an,'o');
30  xlabel('Harmonic Number');
31  ylabel('Magnitude');
32  title('Fourier Series Magnitude')
33
34  //Plotting fourier series phase
35  subplot(2,2,3)
36  plot(n,ph_an,'o')
37  xlabel('Harmonic Number')
38  ylabel('Phase (deg)')
39  title('Fourier Series Phase'),
```

Experiment: 13

Causal and Non-Causal

Scilab code Solution 13.13 To find whether the system is causal or non causal

```
1
2 // VERSION: Scilab: 5.4.1
3 // OS: windows 7
4
5 //CAPTION: To find whether the system is causal or
       non−causal
6
7 clc;
8 clear all;
9 close;
10 // %Properties of DT Systems(Causality)
11 //%y(n)=x(−n);
12
13 x1=input('Enter input sequence x1:');
14 n1=input('Enter upper limit n1:');
15 n2=input('Enter lower limit n2:');
16 flag=0;
17 for n=n1:n2
18    arg=-n;
19    if arg>n;
```

```
20          flag =1;
21  end ;
22  end ;
23   if ( flag ==1)
24      disp ( ' system is causal ' ) ;
25  else
26      disp ( ' system is non−causal ' ) ;
27      end
28
29  /// Sample inputs
30  // Enter input sequence x1 :[1  2  −1  3  −4  5  −5]
31  // Enter upper limit n1 :2
32  // Enter lower limit n2:−2
33
34  // output :
35  // system is non−causal
```

Experiment: 14

Sampling Theorem

Scilab code Solution 14.14 To find the sampling theoram

```
1
2  // VERSION:  Scilab:  5.4.1
3  // OS:  windows  7
4
5  //CAPTION:  To  find  the  sampling  theoram
6
7  clc;
8  T=0.04;
9  t=0:0.0005:0.02;
10 f = 1/T;
11 n1=0:40;
12 size(n1)
13 xa_t=sin(2*%pi*2*t/T);
14 subplot(2,2,1);
15 plot2d3(200*t,xa_t);
16 title('Verification  of  sampling  theorem');
17 title('Continuous  signal');
18 xlabel('t');
19 ylabel('x(t)');
```
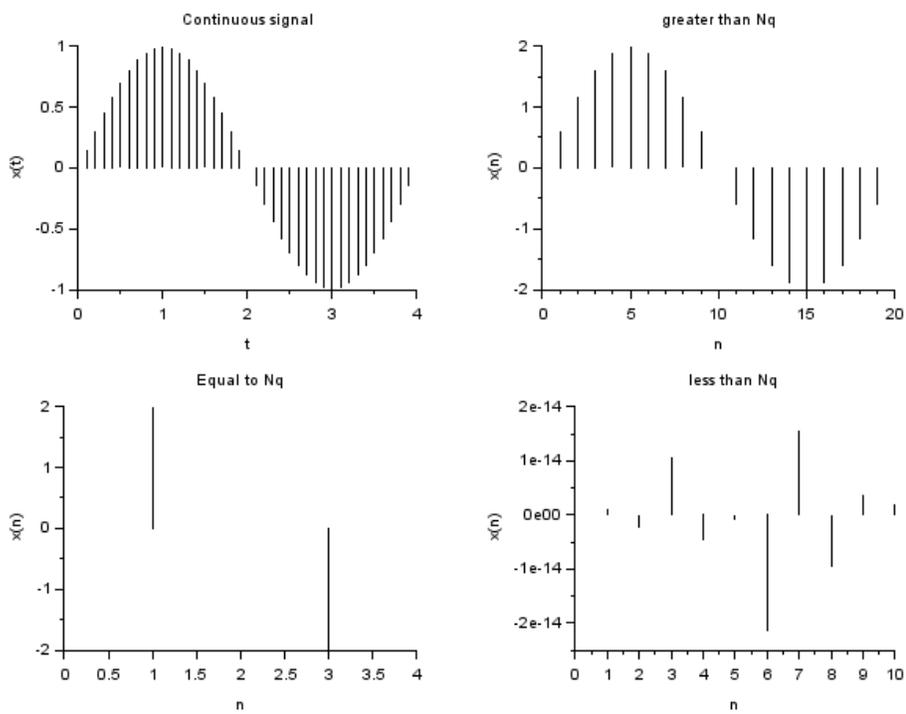
Figure 14.1: To find the sampling theoram

```
20
21  //greater  than  nyquist  rate
22  ts1 =0.002; //>niq  rate
23  n =0:20;
24  x_ts1 =2* sin (2*% pi *n* ts1/T );
25  subplot (2 ,2 ,2) ;
26  plot2d3 (n , x_ts1 );
27  title ('greater  than  Nq ');
28  xlabel ('n ');
29  ylabel ('x(n) ');
30
31  //Equal  to  nyquist  rate
32
33  ts2 =0.01; // niq  rate
34  n =0:4;
35  x_ts2 =2* sin (2*% pi *n* ts2/T );
36  subplot (2 ,2 ,3) ;
37  plot2d3 (n , x_ts2 );
38  title ('Equal  to  Nq ');
39  xlabel ('n ');
40  ylabel ('x(n) ');
41
42  //less  than  nyquist  rate
43  ts3 =0.1; //<niq  rate
44  n =0:10;
45  x_ts3 =2* sin (2*% pi *n* ts3/T );
46  subplot (2 ,2 ,4) ;
47  plot2d3 (n , x_ts3 );
48  title ('less  than  Nq ');
49  xlabel ('n ');
50  ylabel ('x(n) ');
```

Experiment: 15

Sampling rate Conversion

Scilab code Solution 15.15 Sampling Rate Conversion

```
1  // VERSION:  Scilab:  5.4.1
2  // OS:  windows  7
3
4  //CAPTION:  Sampling  Rate  Conversion
5
6  //Down  Sampling  (or  Decimation):
7
8  xn = input(' Enter  the  number  of  samples  xn:');[1 2
        3 4 5 6 8 ]
9  N=length(xn);
10 n=0:1:N-1;
11 D=3;
12 xDn=xn(1:D:N);
13 n1=1:1:N/D;
14 //figure;
15 disp(xDn,'the  downsampling  (or)  Decimation  for  D = 3
        is:———->')
16
17 //Up  Sampling  (or  Interpolation)
18 yn = input(' Enter  the  number  of  samples  yn:')///[1
        -2  3  4  8  9  10  44]
```

```scilab
19  N = length ( yn )
20  n = 0 : 1 : N - 1
21  I = 2
22  xIn = [ zeros ( 1 , I * N ) ]
23  n1 = 1 : 1 : N * I
24  j = 1 : I : I * N
25  xIn ( j ) = yn
26  disp ( xIn , 'The  upsampling  ( or )  interpolation  for  I =
        2  is:-----> ')
27
28
29  ///Sample  Input
30
31  //  Enter  the  number  of  samples  xn : [1  2  3  4  5  6  8  ]
32  // Enter  the  number  of  samples  yn : [1  -2  3  4  8  9  10
      44]
33
34
35  //  Output  Data
36  /////////***************************////////////
37
38  // the  downsampling  ( or )  Decimation  for  D = 3  is
       :----->
39
40  // 1 .      4 .       8 .
41
42  /////////**************************////////////
43
44  ///The  upsampling  ( or )  interpolation  for  I = 2  is
       :----->
45
46  // column  1  to  9
47
48   //     1 .      0 .   -  2 .      0 .      3 .      0 .      4 .      0 .
            8 .
49
50   //            column  10  to  16
51
```

```
52   //    0.     9.     0.     10.     0.     44.     0.
```