# Scilab Manual for Digital Signal Processing by Prof Tirthadip Sinha Electronics Engineering Haldia Institute of Technology[1]

Solutions provided by
Mr TIRTHADIP SINHA
Electronics Engineering
MAKAUT / Haldia Institute of Technology

February 19, 2026

[1]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 3

1 Sampled sinusoidal signal, various sequences and different arithmetic operations 5

2 Convolution of two sequences using graphical methods and using commands 22

3 verification of the properties of convolution 30

4 Z-transform of various sequences 38

# List of Experiments


Solution 1.1 1 . . . . . 5
Solution 1.2 2 . . . . . 7
Solution 1.3 3 . . . . . 10
Solution 1.4 4 . . . . . 10
Solution 1.5 5 . . . . . 12
Solution 1.6 6 . . . . . 14
Solution 1.7 7 . . . . . 17
Solution 1.8 8 . . . . . 17
Solution 1.9 9 . . . . . 19
Solution 2.1 1 . . . . . 22
Solution 2.2 2 . . . . . 24
Solution 2.3 3 . . . . . 26
Solution 3.1 1 . . . . . 30
Solution 3.2 2 . . . . . 32
Solution 3.3 3 . . . . . 34
Solution 4.1 1 . . . . . 38
Solution 4.2 2 . . . . . 41

# List of Figures


1.1 1 . . . . . 6
1.2 2 . . . . . 8
1.3 3 . . . . . 9
1.4 4 . . . . . 11
1.5 5 . . . . . 13
1.6 6 . . . . . 15
1.7 7 . . . . . 16
1.8 8 . . . . . 18
1.9 9 . . . . . 20

2.1 1 . . . . . 23
2.2 2 . . . . . 25
2.3 3 . . . . . 27

3.1 1 . . . . . 31
3.2 2 . . . . . 33
3.3 3 . . . . . 35

4.1 1 . . . . . 39
4.2 2 . . . . . 40

# Experiment: 1

# Sampled sinusoidal signal, various sequences and different arithmetic operations

**Scilab code Solution 1.1** 1

```
1  //Example 1_1
2  //Plot the continuous time signal x(t)=exp(-2t) and
       also its discrete time equivalent signal with a
       sampling period T = 0.2sec
3  clear;
4  clc ;
5  close ;
6  t=0:0.01:2;
7  x1=exp(-2*t);
8  subplot(1,2,1);
9  plot(t,x1);
10 xlabel('t ->');
11 ylabel('x(t) ->');
12 title('Continuous-time Plot of exp (-2t)');
13 n=0:0.2:2;
```

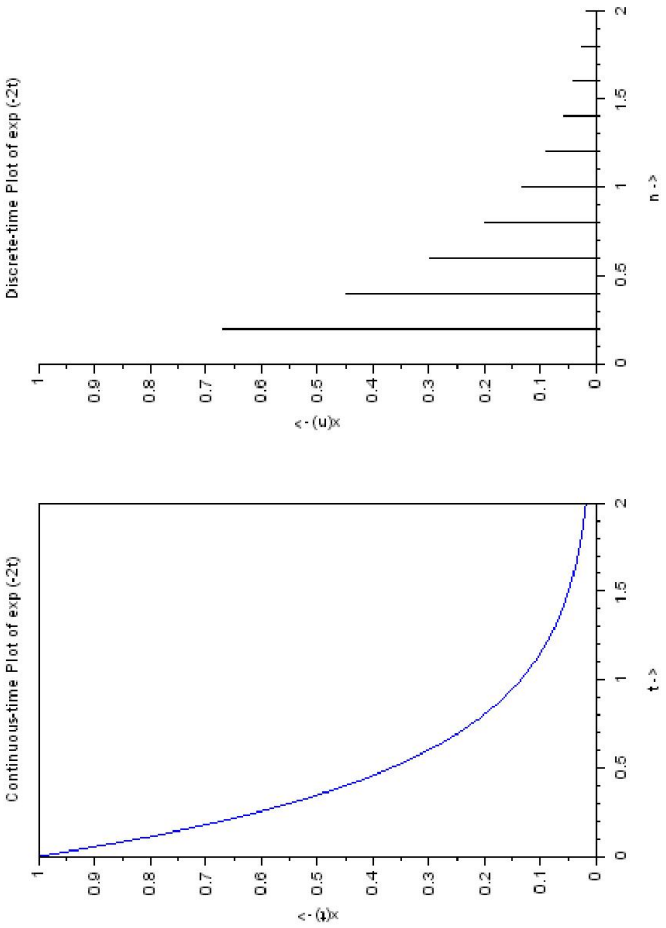
Discrete-time Plot of exp (-2t)
x(n) ->
n ->
Continuous-time Plot of exp (-2t)
x(t) ->
t ->
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.5
1
1.5
2

Figure 1.1: 1

```
14 x2=exp(-2*n);
15 subplot(1,2,2);
16 plot2d3(n,x2);
17 xlabel('n ->');
18 ylabel('x(n) ->');
19 title('Discrete-time Plot of exp (-2t)');
```

**Scilab code Solution 1.2** 2

```
1  //Example 1_2
2  //Plot the continuous time signal x=sin(10*t) and
     also its discrete time equivalent signal with a
     sampling period T = 0.2sec
3  clear;
4  clc ;
5  close ;
6  t=0:0.01:2;
7  x1=sin(10*t);
8  subplot(1,2,1);
9  plot(t,x1);
10 xlabel('t ->');
11 ylabel('x(t) ->');
12 title('Continuous-time Plot of sin (10*t)');
13 n=0:0.2:2;
14 x2=sin(10*n);
15 subplot(1,2,2);
16 plot2d3(n,x2);
17 xlabel('n ->');
18 ylabel('x(n) ->');
19 title('Discrete-time Plot of sin (10*t)');
```

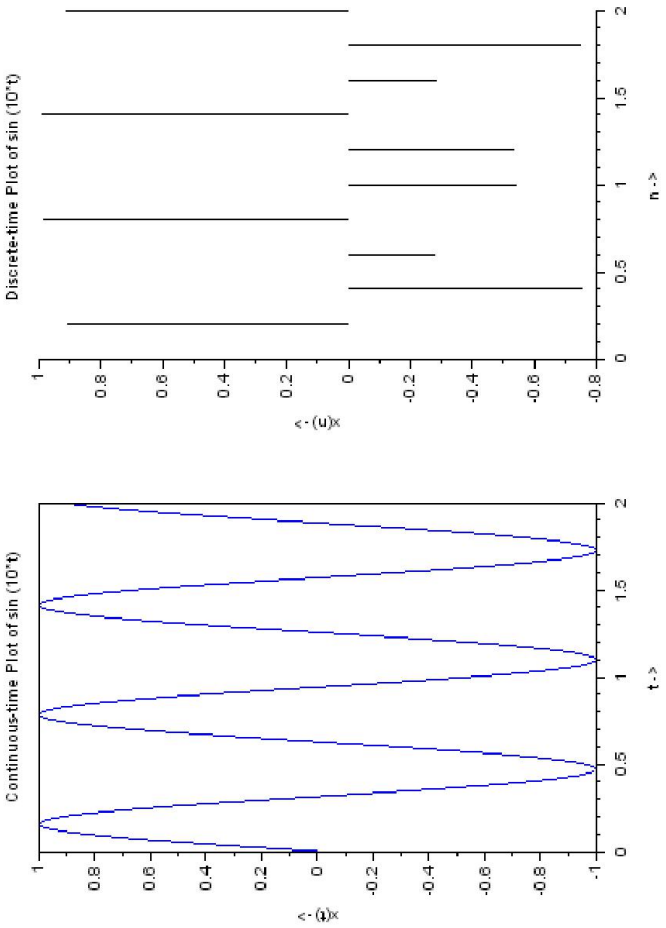
Discrete-time Plot of sin (10*t)
x(n) ->
n ->
Continuous-time Plot of sin (10*t)
x(t) ->
t ->
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.5
1
1.5
2

Figure 1.2: 2

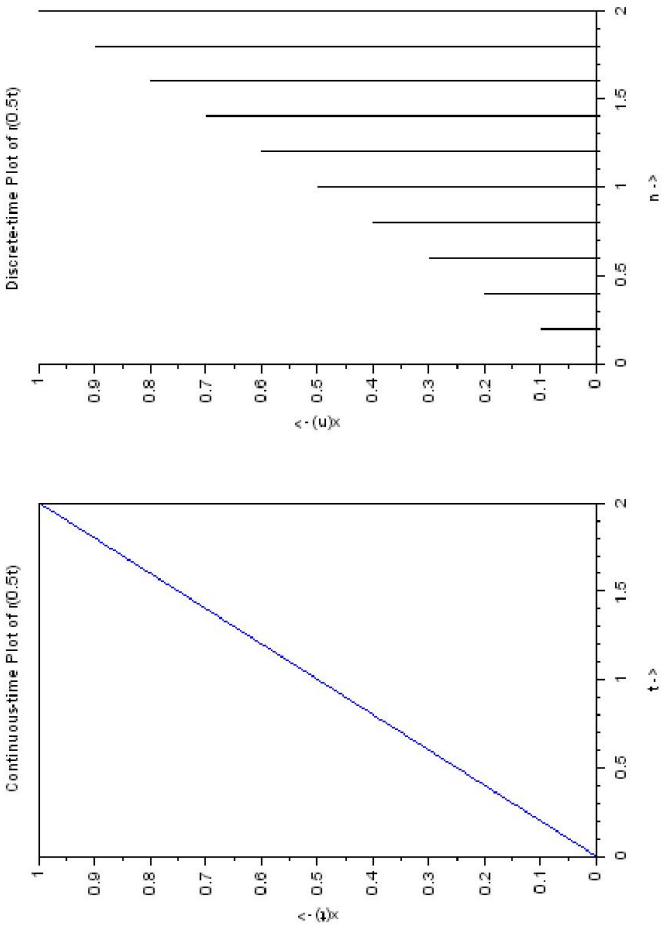
Discrete-time Plot of r(0.5t)
x(n) ->
n ->
Continuous-time Plot of r(0.5t)
x(t) ->
t ->
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.5
1
1.5
2

Figure 1.3: 3

**Scilab code Solution 1.3** 3

```
1 //Example 1_3: Plot the continuous time signal x(t)=
    r(0.5*t) and also its discrete time equivalent
    signal with a sampling period T = 0.2sec
2
3 clc;clear;
4 t=0:0.01:2;
5 x1=0.5*t;
6 subplot(1,2,1);
7 plot(t,x1);
8 xlabel('t ->');
9 ylabel('x(t) ->');
10 title('Continuous-time Plot of r(0.5t)');
11 //zoom_rect([-5 0 5 5]);
12 n=0:0.2:2;
13 x2=0.5*n;
14 subplot(1,2,2);
15 plot2d3(n,x2);
16 xlabel('n ->');
17 ylabel('x(n) ->');
18 title('Discrete-time Plot of r(0.5t)');
```

**Scilab code Solution 1.4** 4

```
1 //Example 1_4: Plot the continuous time signal x(t)=
    rect(2*t) and also its discrete time equivalent
    signal with a sampling period T = 0.01sec
2 clc;
3 clear;
```

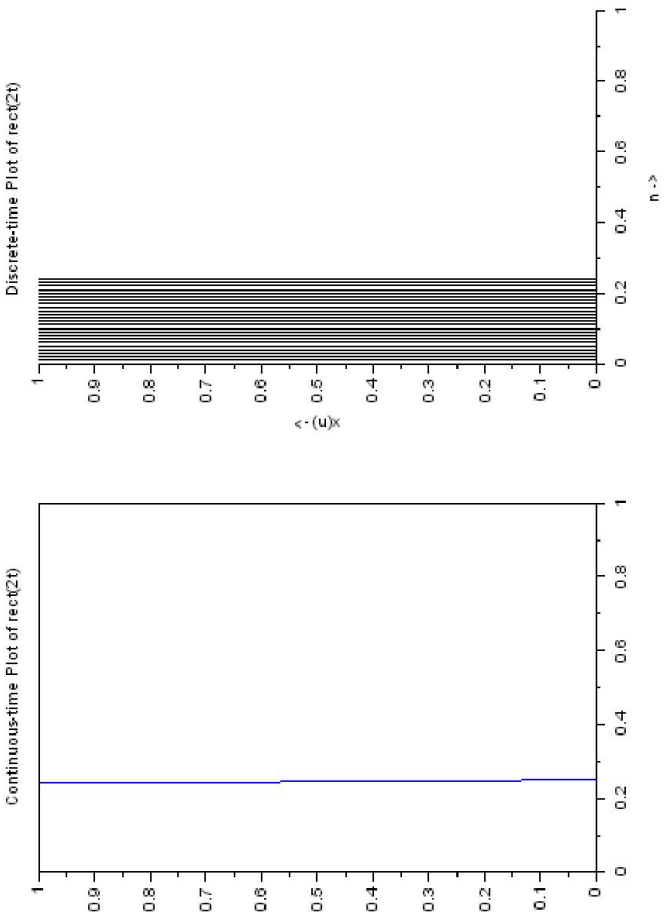
Discrete-time Plot of rect(2t)
x(n) ->
n ->
Continuous-time Plot of rect(2t)

Figure 1.4: 4

```
t=0:0.01:1;
x1=1*(abs(2*t)<0.5);
subplot(1,2,1);
plot(t,x1);
title('Continuous-time Plot of rect(2t)');
n=0:0.01:1;
x2=1*(abs(2*n)<0.5);
subplot(1,2,2);
plot2d3(n,x2);
xlabel('n ->');
ylabel('x(n) ->');
title('Discrete-time Plot of rect(2t)');
```

**Scilab code Solution 1.5** 5

```
//Program 1_5: Plot the continuous time signal x=
   sinc(10*t) and also its discrete time equivalent
   signal with a sampling period T = 0.1sec
clear;
clc ;
close ;
t=-2:0.01:2;
x1=sinc(10*t);
subplot(1,2,1);
plot(t,x1);
xlabel('t ->');
ylabel('x(t) ->');
title('Continuous-time Plot of sinc (10*t)');
n=-2:0.1:2;
x2=sinc(10*n);
subplot(1,2,2);
plot2d3(n,x2);
xlabel('n ->');
```

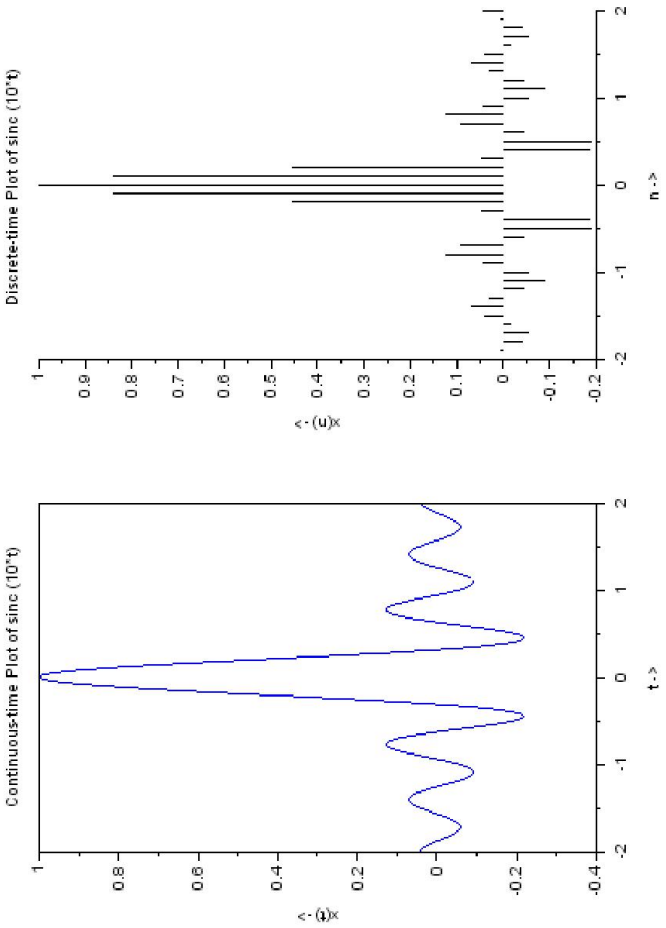
Discrete-time Plot of sinc (10*t)
x(n) ->
n ->
Continuous-time Plot of sinc (10*t)
x(t) ->
t ->

Figure 1.5: 5

```
17 ylabel('x(n) ->');
18 title('Discrete-time Plot of sinc (10*t)');
```

**Scilab code Solution 1.6** 6

```
1 //program 1_6:Write a Scilab program to verify the
    Time Shifting arithmetic operations of the
    discrete-time sequence x[n]=[1 2 5 3 6]
2
3 clear;
4 clc;
5 close;
6 n=-3:3;
7 x=[0 1 2 5 3 6 0];
8 m=input('Enter the value of m = ');
9 subplot(3,1,1);
10 plot2d3(n,x);
11 xlabel('n ->');
12 ylabel('x(n) ->');
13 title('Discrete-time sequences');
14 subplot(3,1,2);
15 plot2d3(n+(-m),x);
16 xlabel('n ->');
17 ylabel('x(n+m) ->');
18 title('Advance-time sequences (Left shift)');
19 subplot(3,1,3);
20 plot2d3(n-(-m),x);
21 xlabel('n ->');
22 ylabel('x(n-m) ->');
23 title('Delay-time sequences (Right shift)');
```

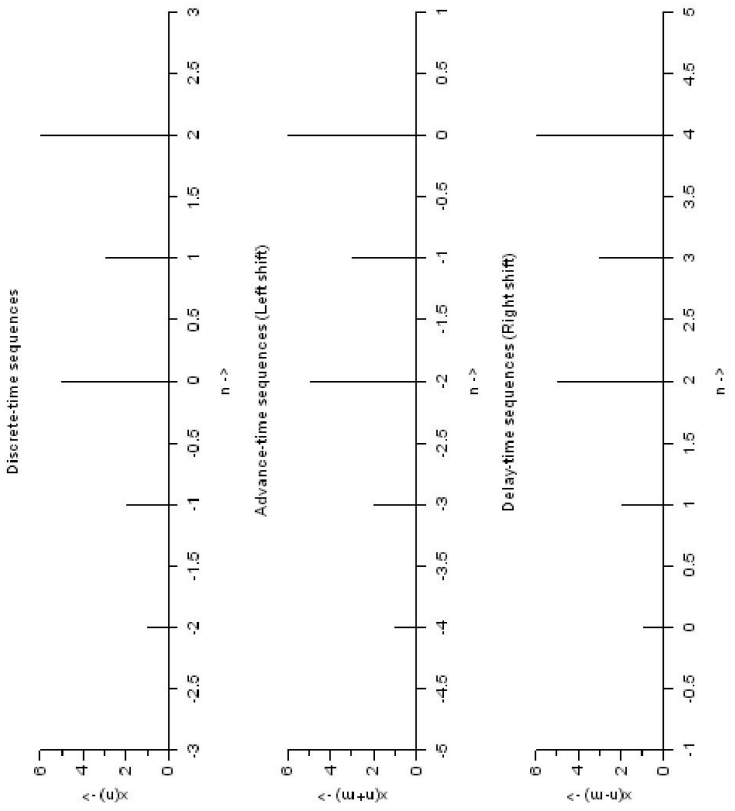
Discrete-time sequences
x(n) ->
n ->
Advance-time sequences (Left shift)
x(n+m) ->
n ->
Delay-time sequences (Right shift)
x(n-m) ->
n ->

Figure 1.6: 6

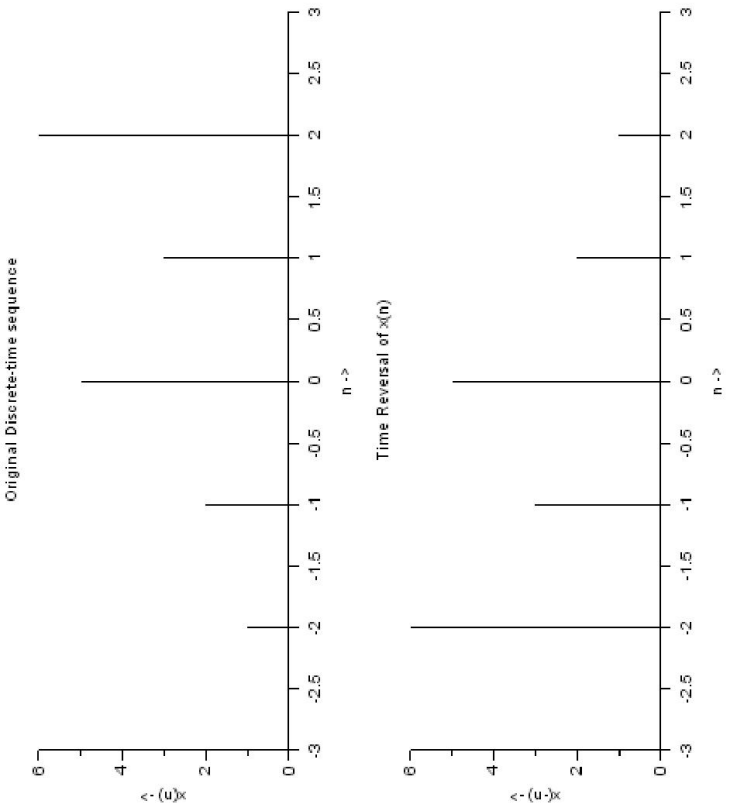
Original Discrete-time sequence
x(n) <- (u)x
n ->
Time Reversal of x(n)
x(-n) <- (u-)x
n ->
6
4
2
0
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3

Figure 1.7: 7

**Scilab code Solution 1.7** 7

```
1 //Program 1_7: Write a Scilab program to verify the
    Time Reversal/Folding arithmetic operations of
    the discrete-time sequence x[n]=[1 2 5 3 6]
2
3 clear;
4 clc;
5 close;
6 n=-3:3;
7 x=[0 1 2 5 3 6 0];
8 subplot(2,1,1);
9 plot2d3(n,x);
10 xlabel('n ->');
11 ylabel('x(n) ->');
12 title('Original Discrete-time sequence');
13 subplot(2,1,2);
14 plot2d3(-n,x);
15 xlabel('n ->');
16 ylabel('x(-n) ->');
17 title('Time Reversal of x(n)');
```

**Scilab code Solution 1.8** 8

```
1 //Program 1_8: Write a Scilab program to verify the
    time scaling operations of the discrete-time
    sequence x[n]=[1 2 3 4 5 6]
2
3 clear;
4 clc;
```

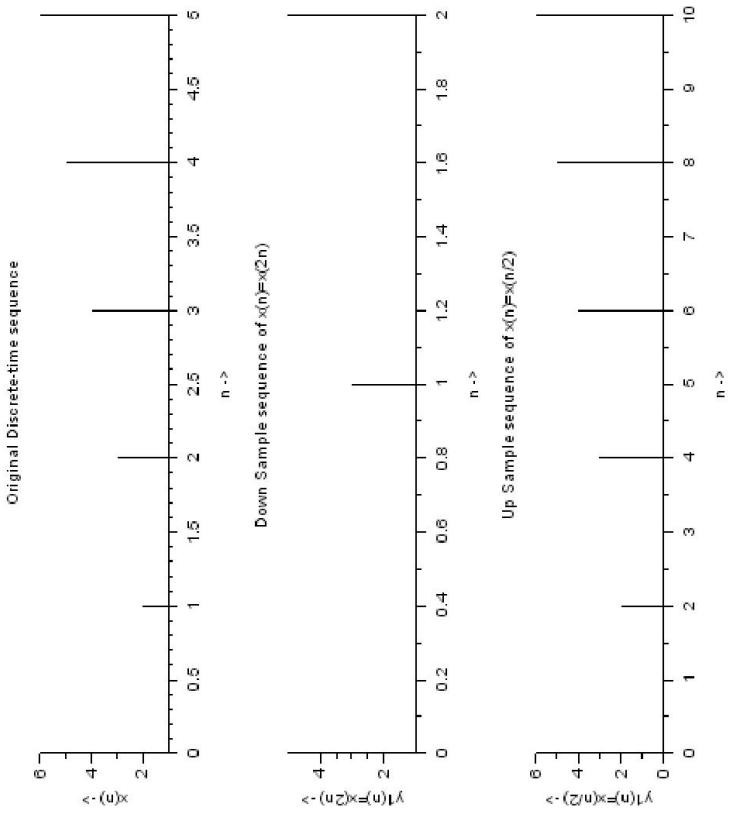
Original Discrete-time sequence
x(n) ->
n ->
Down Sample sequence of x(n)=x(2n)
y1(n)=x(2n) ->
n ->
Up Sample sequence of x(n)=x(n/2)
y1(n)=x(n/2) ->
n ->

Figure 1.8: 8

```
close;
n=0:5;
x=[1 2 3 4 5 6];
subplot(3,1,1);
plot2d3(n,x)
xlabel('n ->');
ylabel('x(n) ->');
title('Original Discrete-time sequence');
a=2;
m=length(n);
//Down Sample
n1=0:(m/2)-1;
y1=[x(1), x(1+(a)), x(1+2*(a))];
subplot(3,1,2)
plot2d3(n1,y1);
xlabel('n ->');
ylabel('y1(n)=x(2n) ->');
title('Down Sample sequence of x(n)=x(2n)');
//Up Sample
n2=0:2*(m-1);
y2=[x(1), 0, x(2), 0, x(3), 0, x(4), 0, x(5), 0, x
   (6)];
subplot(3,1,3);
plot2d3(n2,y2)
xlabel('n ->');
ylabel('y1(n)=x(n/2) ->');
title('Up Sample sequence of x(n)=x(n/2)');
```

**Scilab code Solution 1.9** 9

```
//program 1_9: Write a Scilab program to verify the
   Addition or Multiplication operations of the
   discrete-time sequences x1[n]=[0 1 2 3 4 5 6] and
```

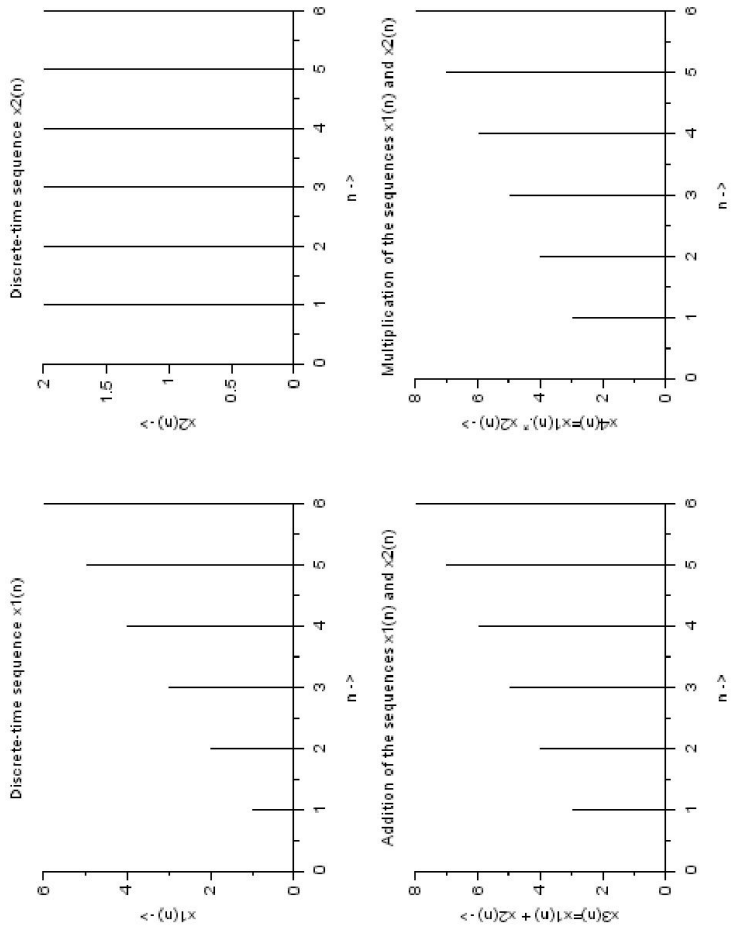
Discrete-time sequence x2(n)
x2(n) ->
n ->
Discrete-time sequence x1(n)
x1(n) ->
n ->
Multiplication of the sequences x1(n) and x2(n)
x4(n)=x1(n).* x2(n) ->
n ->
Addition of the sequences x1(n) and x2(n)
x3(n)=x1(n) + x2(n) ->
n ->

Figure 1.9: 9

```
    x2[n]=[0  2   2   2   2   2   2]
2
3  clear;
4  clc;
5  close;
6  n=0:6;
7  x1=[0 1 2 3 4 5 6];
8  x2=[0 2 2 2 2 2 2];
9  subplot(2,2,1);
10 plot2d3(n,x1);
11 xlabel('n ->');
12 ylabel('x1(n) ->');
13 title('Discrete-time sequence x1(n)');
14 subplot(2,2,2);
15 plot2d3(n,x2);
16 xlabel('n ->');
17 ylabel('x2(n) ->');
18 title('Discrete-time sequence x2(n)');
19 x3=x1+x2;
20 subplot(2,2,3);
21 plot2d3(n,x3);
22 xlabel('n ->');
23 ylabel('x3(n)=x1(n) + x2(n) ->');
24 title('Addition of the sequences x1(n) and x2(n)');
25 x4=x1.*x2;
26 subplot(2,2,4);
27 plot2d3(n,x3);
28 xlabel('n ->');
29 ylabel('x4(n)=x1(n).* x2(n) ->');
30 title('Multiplication of the sequences x1(n) and x2(
   n)');
```

# Experiment: 2

# Convolution of two sequences using graphical methods and using commands

**Scilab code Solution 2.1** 1

```
1  //Part1_Convolution of two sequences x1[n] & x2[n]
2  clear;
3  clc;
4  close;
5  n1=0:4;
6  x1=[1 2 0.5 1 2];
7  subplot(2,2,1);
8  plot2d3(n1,x1);
9  xlabel('n ->');
10 ylabel('x1[n] ->');
11 title('Discrete-time sequences x1[n]');
12 n2=-1:3;
13 x2=[1 2 1 -1 2];
14 subplot(2,2,2);
15 plot2d3(n2,x2);
```

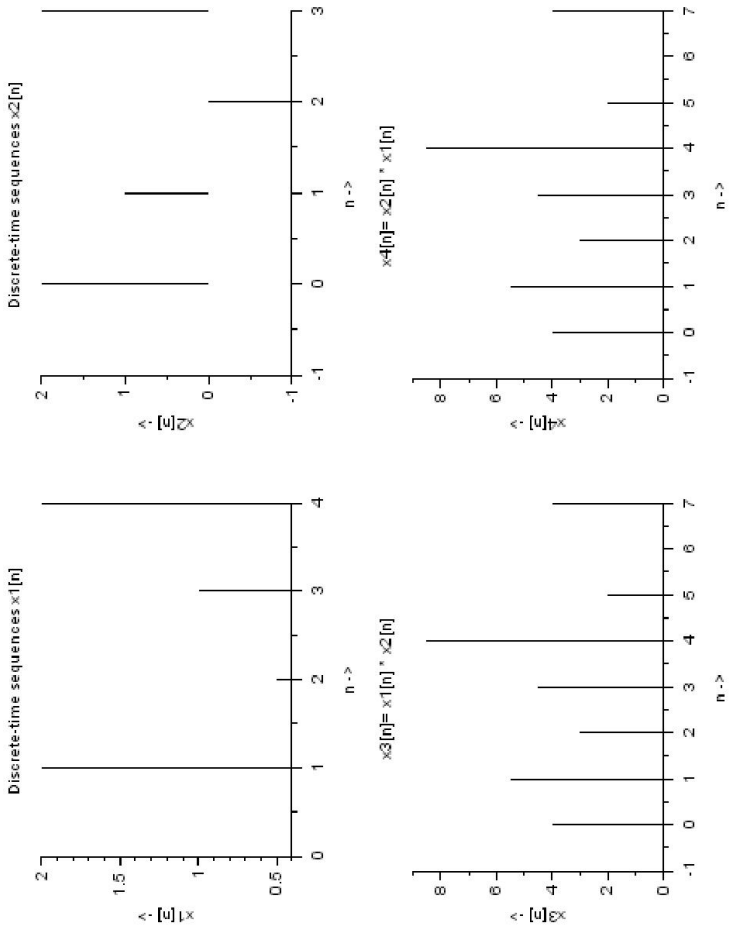
Discrete-time sequences x1[n]
x1[n] ->
n ->
Discrete-time sequences x2[n]
x2[n] ->
n ->
x3[n]= x1[n] * x2[n]
x3[n] ->
n ->
x4[n]= x2[n] * x1[n]
x4[n] ->
n ->

Figure 2.1: 1

```
16 xlabel('n ->');
17 ylabel('x2[n] ->');
18 title('Discrete-time sequences x2[n]');
19 //Verification of Commutative Property
20 n3=-1:7;
21 x3=convol(x1,x2)
22 x4=convol(x2,x1)
23 subplot(2,2,3);
24 plot2d3(n3,x3)
25 xlabel('n ->');
26 ylabel('x3[n] ->');
27 title('x3[n]= x1[n] * x2[n]');
28 subplot(2,2,4);
29 plot2d3(n3, x4);
30 xlabel('n ->');
31 ylabel('x4[n] ->');
32 title('x4[n]= x2[n] * x1[n]');
```

**Scilab code Solution 2.2** 2

```
1 //Program 2_Verifivation of Distributive Property of
    Convolution
2 clear;
3 clc;
4 close;
5 n1=0:4;
6 x1=[1 2 0.5 1 2];
7 subplot(2,3,1);
8 plot2d3(n1,x1);
9 xlabel('n ->');
10 ylabel('x1[n] ->');
11 title('Discrete-time sequences x1[n]');
12 n2=-1:3;
```

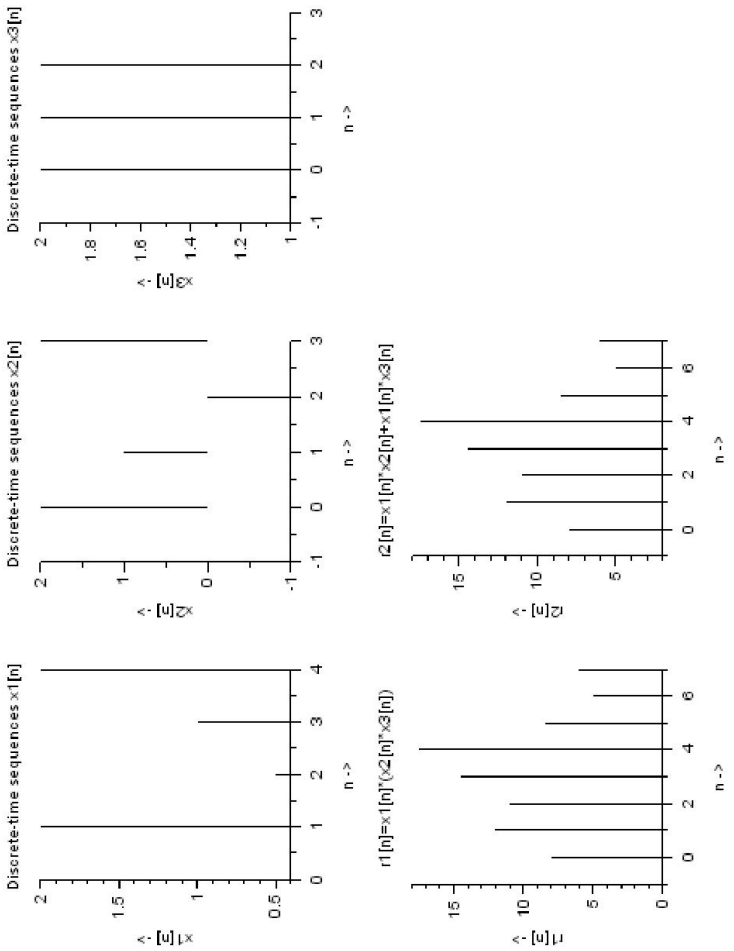
Discrete-time sequences x1[n]
x1[n] ->
n ->
Discrete-time sequences x2[n]
x2[n] ->
n ->
Discrete-time sequences x3[n]
x3[n] ->
n ->
r1[n]=x1[n]*(x2[n]*x3[n])
r1[n] ->
n ->
r2[n]=x1[n]*x2[n]+x1[n]*x3[n]
r2[n] ->
n ->

Figure 2.2: 2

```
13 x2=[1 2 1 -1 2];
14 subplot(2,3,2);
15 plot2d3(n2,x2);
16 xlabel('n ->');
17 ylabel('x2[n] ->');
18 title('Discrete-time sequences x2[n]');
19 n3=-1:3;
20 x3=[1 2 2 2 1];
21 subplot(2,3,3);
22 plot2d3(n3,x3);
23 xlabel('n ->');
24 ylabel('x3[n] ->');
25 title('Discrete-time sequences x3[n]');
26 //Verification of Distributive Property
27 s1=-1:7;
28 x4=x2+x3;
29 r1=convol(x1,x4);
30 subplot(2,3,4);
31 plot2d3(s1,r1);
32 xlabel('n ->');
33 ylabel('r1[n] ->');
34 title('r1[n]=x1[n]*(x2[n]*x3[n])');
35 x4=convol(x1,x2);
36 x5=convol(x1,x3);
37 r2=x4+x5;
38 subplot(2,3,5);
39 plot2d3(s1,r2);
40 xlabel('n ->');
41 ylabel('r2[n] ->');
42 title('r2[n]=x1[n]*x2[n]+x1[n]*x3[n]');
```

**Scilab code Solution 2.3** 3

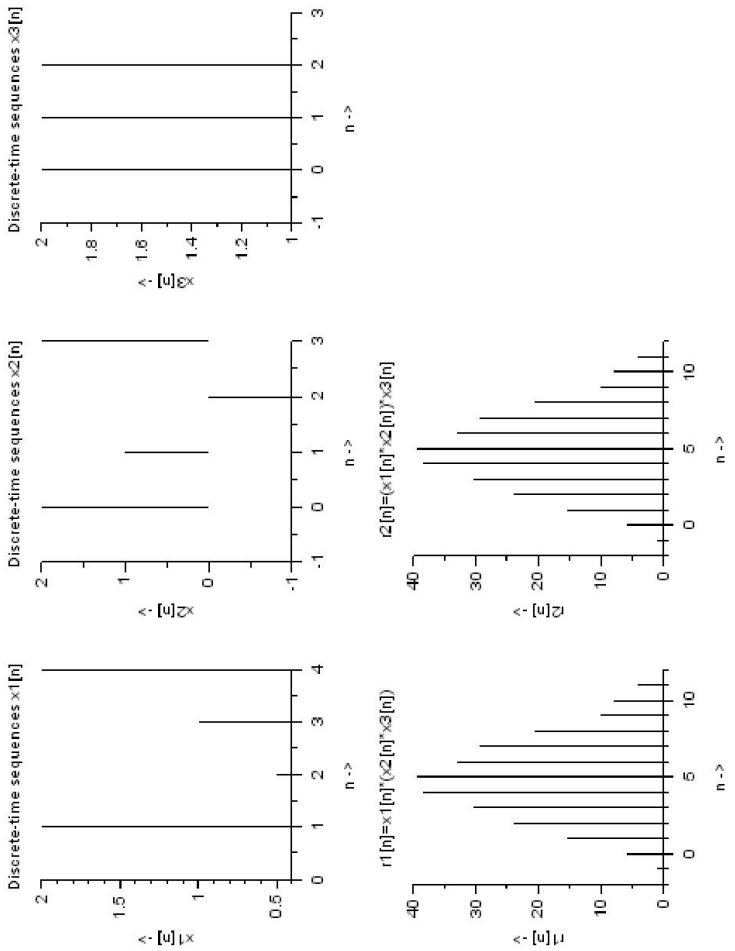
Discrete-time sequences x1[n]
x1[n] ->
n ->
Discrete-time sequences x2[n]
x2[n] ->
n ->
Discrete-time sequences x3[n]
x3[n] ->
n ->
r1[n]=x1[n]*(x2[n]*x3[n])
r1[n] ->
n ->
r2[n]=(x1[n]*x2[n])*x3[n]
r2[n] ->
n ->

Figure 2.3: 3

```
1  //Program 3_Verifivation of Associative Property of
     Convolution
2  clear;
3  clc;
4  close;
5  n1=0:4;
6  x1=[1 2 0.5 1 2];
7  subplot(2,3,1);
8  plot2d3(n1,x1);
9  xlabel('n ->');
10 ylabel('x1[n] ->');
11 title('Discrete-time sequences x1[n]');
12 n2=-1:3;
13 x2=[1 2 1 -1 2];
14 subplot(2,3,2);
15 plot2d3(n2,x2);
16 xlabel('n ->');
17 ylabel('x2[n] ->');
18 title('Discrete-time sequences x2[n]');
19 n3=-1:3;
20 x3=[1 2 2 2 1];
21 subplot(2,3,3);
22 plot2d3(n3,x3);
23 xlabel('n ->');
24 ylabel('x3[n] ->');
25 title('Discrete-time sequences x3[n]');
26 //Verification of Associative Property
27 s1=-1:11;
28 r1=convol(x1,convol(x2,x3));
29 subplot(2,3,4);
30 plot2d3(s1,r1);
31 xlabel('n ->');
32 ylabel('r1[n] ->');
33 title('r1[n]=x1[n]*(x2[n]*x3[n])');
34 r2=convol(convol(x1,x2), x3);
35 subplot(2,3,5);
36 plot2d3(s1,r2);
37 xlabel('n ->');
```

```
38 ylabel('r2[n] ->');
39 title('r2[n]=(x1[n]*x2[n])*x3[n]');
```

# Experiment: 3

# verification of the properties of convolution

**Scilab code Solution 3.1** 1

```
1  //Part1_Convolution of two sequences x1[n] & x2[n]
2  clear;
3  clc;
4  close;
5  n1=0:4;
6  x1=[1 2 0.5 1 2];
7  subplot(2,2,1);
8  plot2d3(n1,x1);
9  xlabel('n ->');
10 ylabel('x1[n] ->');
11 title('Discrete-time sequences x1[n]');
12 n2=-1:3;
13 x2=[1 2 1 -1 2];
14 subplot(2,2,2);
15 plot2d3(n2,x2);
16 xlabel('n ->');
17 ylabel('x2[n] ->');
```

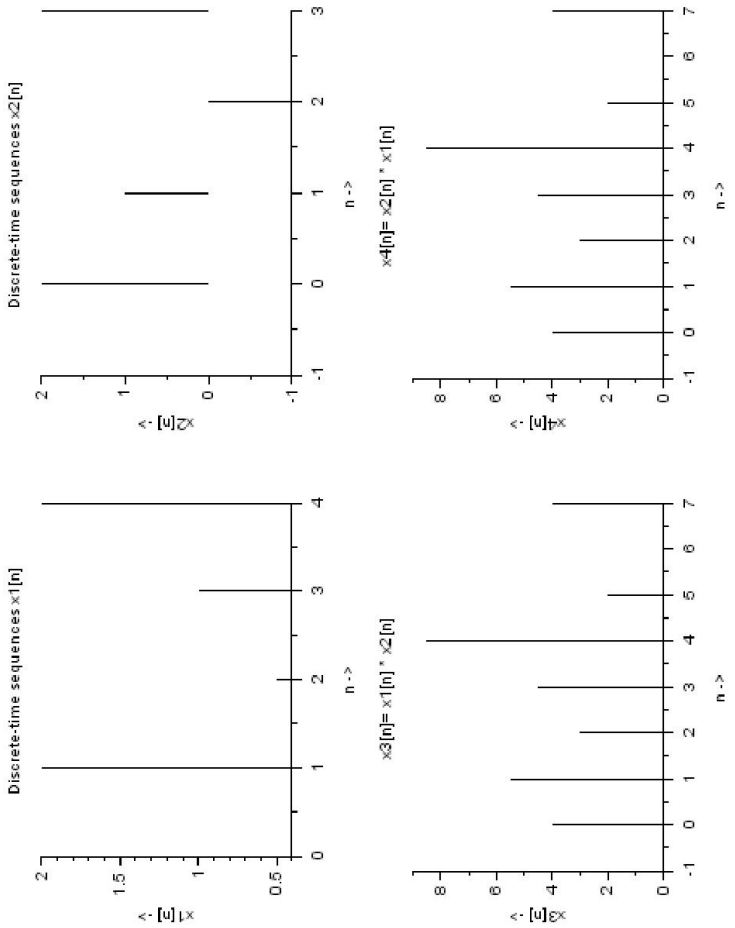
Discrete-time sequences x1[n]
x1[n] ->
n ->
Discrete-time sequences x2[n]
x2[n] ->
n ->
x3[n]= x1[n] * x2[n]
x3[n] ->
n ->
x4[n]= x2[n] * x1[n]
x4[n] ->
n ->

Figure 3.1: 1

```
18 title('Discrete-time sequences x2[n]');
19 //Verification of Commutative Property
20 n3=-1:7;
21 x3=convol(x1,x2)
22 x4=convol(x2,x1)
23 subplot(2,2,3);
24 plot2d3(n3,x3)
25 xlabel('n ->');
26 ylabel('x3[n] ->');
27 title('x3[n]= x1[n] * x2[n]');
28 subplot(2,2,4);
29 plot2d3(n3, x4);
30 xlabel('n ->');
31 ylabel('x4[n] ->');
32 title('x4[n]= x2[n] * x1[n]');
```

**Scilab code Solution 3.2** 2

```
1 //Program 2_Verifivation of Distributive Property of
    Convolution
2 clear;
3 clc;
4 close;
5 n1=0:4;
6 x1=[1 2 0.5 1 2];
7 subplot(2,3,1);
8 plot2d3(n1,x1);
9 xlabel('n ->');
10 ylabel('x1[n] ->');
11 title('Discrete-time sequences x1[n]');
12 n2=-1:3;
13 x2=[1 2 1 -1 2];
14 subplot(2,3,2);
```

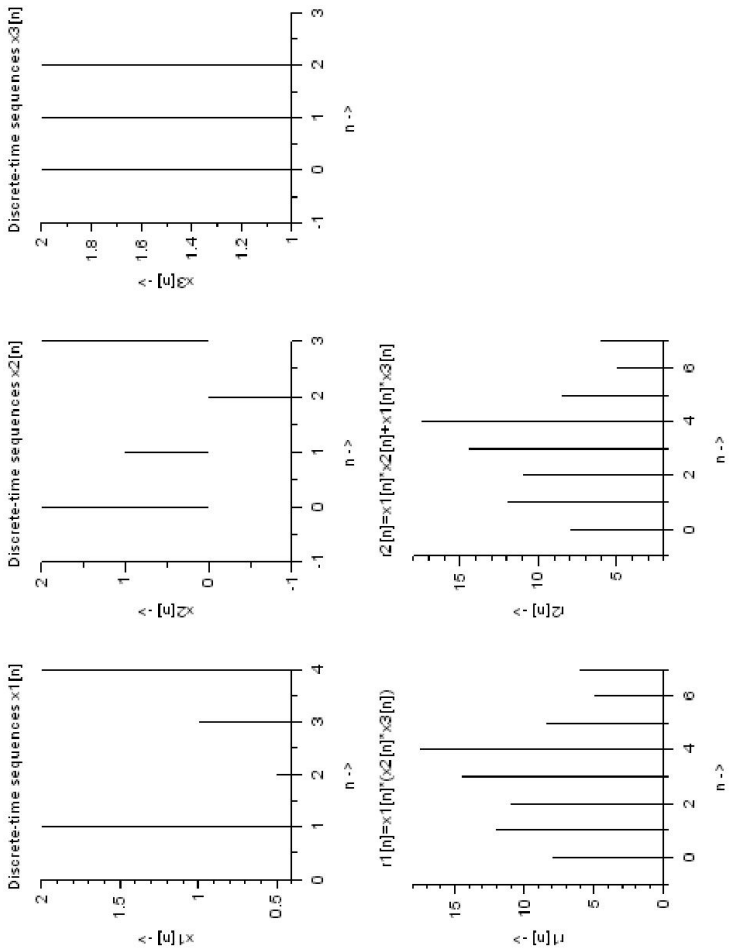
Discrete-time sequences x1[n]
x1[n] ->
n ->
Discrete-time sequences x2[n]
x2[n] ->
n ->
Discrete-time sequences x3[n]
x3[n] ->
n ->
r1[n]=x1[n]*(x2[n]*x3[n])
r1[n] ->
n ->
r2[n]=x1[n]*x2[n]+x1[n]*x3[n]
r2[n] ->
n ->

Figure 3.2: 2

```
15 plot2d3(n2,x2);
16 xlabel('n ->');
17 ylabel('x2[n] ->');
18 title('Discrete-time sequences x2[n]');
19 n3=-1:3;
20 x3=[1 2 2 2 1];
21 subplot(2,3,3);
22 plot2d3(n3,x3);
23 xlabel('n ->');
24 ylabel('x3[n] ->');
25 title('Discrete-time sequences x3[n]');
26 //Verification of Distributive Property
27 s1=-1:7;
28 x4=x2+x3;
29 r1=convol(x1,x4);
30 subplot(2,3,4);
31 plot2d3(s1,r1);
32 xlabel('n ->');
33 ylabel('r1[n] ->');
34 title('r1[n]=x1[n]*(x2[n]*x3[n])');
35 x4=convol(x1,x2);
36 x5=convol(x1,x3);
37 r2=x4+x5;
38 subplot(2,3,5);
39 plot2d3(s1,r2);
40 xlabel('n ->');
41 ylabel('r2[n] ->');
42 title('r2[n]=x1[n]*x2[n]+x1[n]*x3[n]');
```

**Scilab code Solution 3.3** 3

```
1 //Program 3_Verifivation of Associative Property of
    Convolution
```

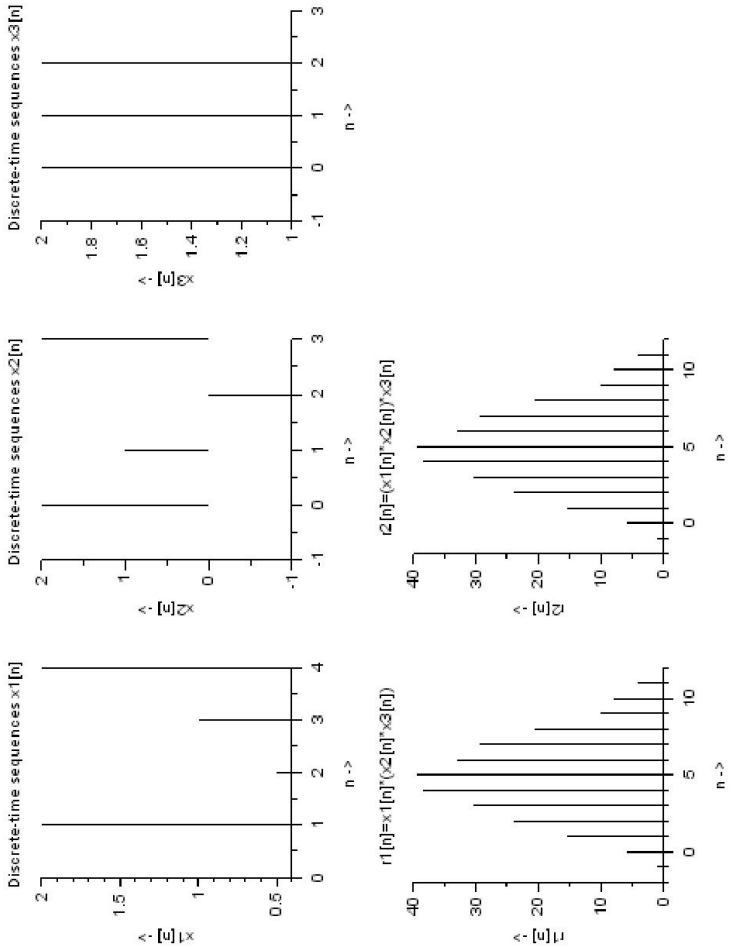
Discrete-time sequences x1[n]
x1[n] ->
n ->
Discrete-time sequences x2[n]
x2[n] ->
n ->
Discrete-time sequences x3[n]
x3[n] ->
n ->
r1[n]=x1[n]*(x2[n]*x3[n])
r1[n] ->
n ->
r2[n]=(x1[n]*x2[n])*x3[n]
r2[n] ->
n ->

Figure 3.3: 3

```
clear;
clc;
close;
n1=0:4;
x1=[1 2 0.5 1 2];
subplot(2,3,1);
plot2d3(n1,x1);
xlabel('n ->');
ylabel('x1[n] ->');
title('Discrete-time sequences x1[n]');
n2=-1:3;
x2=[1 2 1 -1 2];
subplot(2,3,2);
plot2d3(n2,x2);
xlabel('n ->');
ylabel('x2[n] ->');
title('Discrete-time sequences x2[n]');
n3=-1:3;
x3=[1 2 2 2 1];
subplot(2,3,3);
plot2d3(n3,x3);
xlabel('n ->');
ylabel('x3[n] ->');
title('Discrete-time sequences x3[n]');
//Verification of Associative Property
s1=-1:11;
r1=convol(x1,convol(x2,x3));
subplot(2,3,4);
plot2d3(s1,r1);
xlabel('n ->');
ylabel('r1[n] ->');
title('r1[n]=x1[n]*(x2[n]*x3[n])');
r2=convol(convol(x1,x2), x3);
subplot(2,3,5);
plot2d3(s1,r2);
xlabel('n ->');
ylabel('r2[n] ->');
title('r2[n]=(x1[n]*x2[n])*x3[n]');
```

# Experiment: 4

# Z-transform of various sequences

**Scilab code Solution 4.1** 1

```
1 //Program 1:1. Write a Scilab program to determine
    z-transform of the following finite duration
    signal x[n]=[2 4 5 7 0 1]
2
3 x=[2 4 5 7 0 1];
4 p=0;
5 n=-2;
6 for i=1:6
7     p=p+x(i)*%z^(-n);
8     n=n+1;
9 end;
10 disp(p);
```

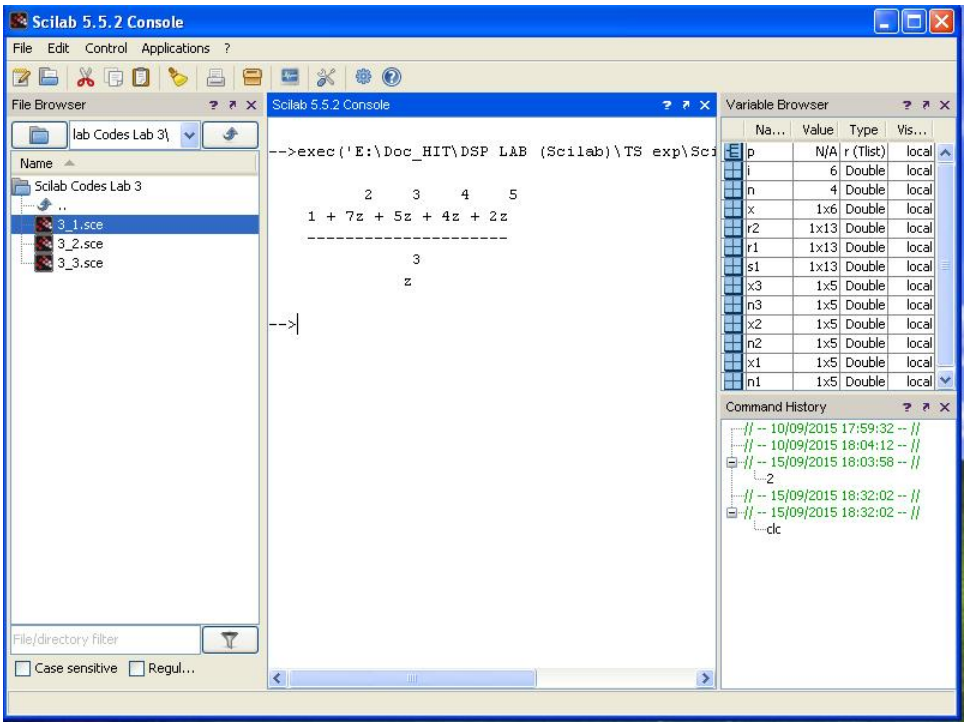
Scilab 5.5.2 Console
File Edit Control Applications ?
File Browser
lab Codes Lab 3\
Name
Scilab Codes Lab 3
..
3_1.sce
3_2.sce
3_3.sce
File/directory filter
Case sensitive
Regul...
Scilab 5.5.2 Console
-->exec('E:\Doc_HIT\DSP LAB (Scilab)\TS exp\Sci
2 3 4 5
1 + 7z + 5z + 4z + 2z
3
z
-->
Variable Browser
Na... Value Type Vis...
p N/A r (Tlist) local
i 6 Double local
n 4 Double local
x 1x6 Double local
r2 1x13 Double local
r1 1x13 Double local
s1 1x13 Double local
x3 1x5 Double local
n3 1x5 Double local
x2 1x5 Double local
n2 1x5 Double local
x1 1x5 Double local
n1 1x5 Double local
Command History
// -- 10/09/2015 17:59:32 -- //
// -- 10/09/2015 18:04:12 -- //
// -- 15/09/2015 18:03:58 -- //
2
// -- 15/09/2015 18:32:02 -- //
// -- 15/09/2015 18:32:02 -- //
clc

Figure 4.1: 1

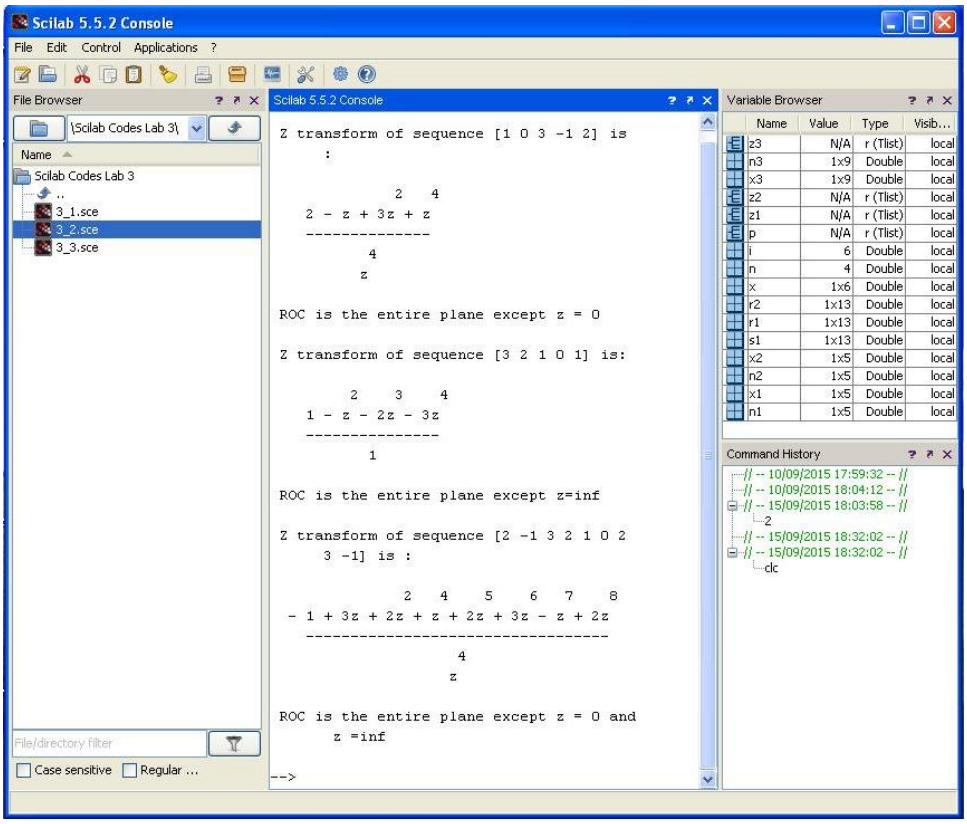
Scilab 5.5.2 Console
File Edit Control Applications ?
File Browser
\Scilab Codes Lab 3\
Name
Scilab Codes Lab 3
..
3_1.sce
3_2.sce
3_3.sce
File/directory filter
Case sensitive
Regular ...
Scilab 5.5.2 Console
Z transform of sequence [1 0 3 -1 2] is
:
2 4
2 - z + 3z + z
4
z
ROC is the entire plane except z = 0
Z transform of sequence [3 2 1 0 1] is:
2 3 4
1 - z - 2z - 3z
1
ROC is the entire plane except z=inf
Z transform of sequence [2 -1 3 2 1 0 2
3 -1] is :
2 4 5 6 7 8
- 1 + 3z + 2z + z + 2z + 3z - z + 2z
4
z
ROC is the entire plane except z = 0 and
z =inf
-->
Variable Browser
Name Value Type Visib...
z3 N/A r (Tlist) local
n3 1x9 Double local
x3 1x9 Double local
z2 N/A r (Tlist) local
z1 N/A r (Tlist) local
p N/A r (Tlist) local
i 6 Double local
n 4 Double local
x 1x6 Double local
r2 1x13 Double local
r1 1x13 Double local
s1 1x13 Double local
x2 1x5 Double local
n2 1x5 Double local
x1 1x5 Double local
n1 1x5 Double local
Command History
// -- 10/09/2015 17:59:32 -- //
// -- 10/09/2015 18:04:12 -- //
// -- 15/09/2015 18:03:58 -- //
2
// -- 15/09/2015 18:32:02 -- //
// -- 15/09/2015 18:32:02 -- //
clc

Figure 4.2: 2

**Scilab code Solution 4.2** 2

```
1 //Program 2_Z transform of Causal, Anticausal and
     Noncausal sequence
2 clear all;
3 clc ;
4 close ;
5 function [za ]= ztransfer ( sequence ,n)
6 za= sequence *(1/ %z)^n'
7 endfunction
8 //z transform of [1 0 3 -1 2]
9 x1 =[1 0 3 -1 2];
10 n1=0:length (x1)-1;
11 z1= ztransfer (x1,n1);
12 disp (z1 ,"Z transform of sequence [1 0 3 -1 2] is :
     ");
13 disp ( 'ROC is the entire plane except z = 0 ' );
14
15 //Z transform of [3 2 1 0 1]
16 x2 =[-3 -2 -1 0 1];
17 n2=-(length(x2)-1):0;
18 z2= ztransfer(x2,n2);
19 disp(z2 ,"Z transform of sequence [3 2 1 0 1] is: ")
     ;
20 disp('ROC is the entire plane except z=inf' );
21
22 //Z transform of [2 -1 3 2 1 0 2 3 -1]
23 x3 =[2 -1 3 2 1 0 2 3 -1];
24 n3= -4:4;
25 z3= ztransfer(x3,n3);
26 disp(z3,"Z transform of sequence [2 -1 3 2 1 0 2 3
     -1] is : ");
27 disp('ROC is the entire plane except z = 0 and z =
     inf' );
```