Scilab Manual for Signals and Systems by Prof Manisha Joshi Electronics Engineering VESIT[1]

Solutions provided by
Vivek Goklani
Electronics Engineering
VESIT

April 25, 2024

[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 Convolution of discrete signal using convol and convolution sum 4

2 To plot pole zero plot for different transfer function 7

3 To obtain Fourier transform of discrete signal 9

4 To find even and odd components of given signal 10

5 To implement representation of elementary signals using scilab 12

6 To study time shift and time advance 14

List of Experiments

Solution 1.1	convolution of discrete sequences	4
Solution 2.2	pole zero plot	7
Solution 3.3	fourier transform	9
Solution 4.4	even and odd signals	10
Solution 5.5	elementary signals	12
Solution 6.6	time shifting	14

Experiment: 1

Convolution of discrete signal using convol and convolution sum

Scilab code Solution 1.1 convolution of discrete sequences

```
1  //Experiment-1
2  //Scilab 5.4.1, Windows 7
3  //convolution using convol command
4  t1=[0:3]
5  n=2*length(t1)-1
6  t2=1:n
7  x=[1 2 3 4]//input signal
8  h=[5 6 7 8]//impulse input
9  y=convol(x,h)//convoluted signal
10 subplot(2,3,1)
11 plot2d3(t1,x,rect=[0,0,10,10])
12 xlabel('t->','fontsize',5)
13 ylabel('x(n)','fontsize',5)
14 title('signal x(n)','fontsize',5)
15 subplot(2,3,2)
16 plot2d3(t1,h,rect=[0,0,10,10])
17 xlabel('t->','fontsize',5)
```

```
18 ylabel('h(n)','fontsize',5)
19 title('signal h(n)','fontsize',5)
20 subplot(2,3,3)
21 plot2d3(t2,y,rect=[0,0,10,100])
22 xlabel('t->','fontsize',5)
23 ylabel('y(n)','fontsize',5)
24 title('convolved signal (convol command)','fontsize'
      ,5)
25 printf("convolved signal using convol command\n")
26 disp(y)
27 printf("\n")
28
29 //verification using convolution sum formula
30 y=[0 0 0 0 0 0 0]
31 t=[1:7]
32 for k=1:4
33     if((2-k)>0) then
34      y(1)=y(1)+x(k)*h(2-k)
35      end
36 end
37 for k=1:4
38     if((3-k)>0) then
39      y(2)=y(2)+x(k)*h(3-k)
40      end
41 end
42 for k=1:4
43     if((4-k)>0) then
44      y(3)=y(3)+x(k)*h(4-k)
45      end
46 end
47 for k=1:4
48     if((5-k)>0) then
49      y(4)=y(4)+x(k)*h(5-k)
50      end
51 end
52 for k=1:4
53     if((6-k)<5) then
54      y(5)=y(5)+x(k)*h(6-k)
```

```
55        end
56  end
57  for k=1:4
58      if((7-k)<5) then
59       y(6)=y(6)+x(k)*h(7-k)
60       end
61  end
62  for k=1:4
63      if((8-k)<5) then
64       y(7)=y(7)+x(k)*h(7-k)
65       end
66  end
67  y(7)=y(7)+4
68  printf("convolved sequence using convolution sum\n")
69  disp(y)
70  subplot(2,3,5)
71  plot2d3(t,y,rect=[0,0,10,100])
72  xlabel('t->','fontsize',5)
73  ylabel('y(n)','fontsize',5)
74  title('convolved signal (convolution sum)','fontsize
      ',5)
```

Experiment: 2

To plot pole zero plot for different transfer function

Scilab code Solution 2.2 pole zero plot

```
1  //Experiment-2
2  //Scilab 5.4.1, Windows 7
3  s=%s
4  n1=[s+3]//transfer function-1
5  d1=[1+3*s;s+5]
6  subplot(2,2,1)
7  h1=syslin('c',n1./d1)
8  plzr(h1)
9  title('(s+3)/(1+3*s)(s+5)','fontsize',5)
10 n2=[s+4]//transfer function-2
11 d2=[1+8*s;s+3]
12 subplot(2,2,2)
13 h2=syslin('c',n2./d2)
14 plzr(h2)//pole zero plot
15 title('(s+4)/(1+8*s)(s+5)','fontsize',5)
16 n3=[s+4]//transfer function-3
17 d3=[s*[2+3*s;s+7]]
18 subplot(2,2,3)
19 h3=syslin('c',n3./d3)//s- domain transfer function
```

```
20 plzr(h3)
21 title('(s+4)/s(2+3*s)(s+7)','fontsize',5)
22 n4=[s^2+4*s+7]//transfer function-4
23 d4=[s^2+2*s+2]
24 subplot(2,2,4)
25 h4=syslin('c',n4./d4)
26 plzr(h4)
27 title('(s^2+4*s+7)/(s^2+2*s+2)','fontsize',5)
```

Experiment: 3

To obtain Fourier transform of discrete signal

Scilab code Solution 3.3 fourier transform

```
1 //Experiment-3
2 //Scilab 5.4.1, Windows 7
3 x=[1 2 3 4]//given sequence
4 y=fft(x,-1)//fast fourier transform
5 printf("fourier transform of sequence\n")
6 disp(y)
```

Experiment: 4

To find even and odd components of given signal

Scilab code Solution 4.4 even and odd signals

```
1  //Experiment-4
2  //Scilab 5.4.1, Windows 7
3  t=[0:0.1:10]//time axis
4  a=exp(%i*t)//given signal
5  b=exp(-%i*t)
6  subplot(2,2,1.5)
7  plot(t,a)
8  title('original signal','fontsize',5)
9  xlabel('Time')
10 ylabel('Amplitude')
11 x=(1/2)*(a+b)//even part of signal
12 y=(1/2)*(a-b)//odd part of signal
13 subplot(2,2,3)
14 plot(t,x)
15 title('even signal','fontsize',5)
16 xlabel('Time')
17 ylabel('Amplitude')
18 subplot(2,2,4)
19 plot(t,y)
```

```
20 title('odd signal','fontsize',5)
21 xlabel('Time')
22 ylabel('Amplitude')
```

Experiment: 5

To implement representation of elementary signals using scilab

Scilab code Solution 5.5 elementary signals

```
1  //Experiment-5
2  //Scilab 5.4.1, Windows 7
3  stacksize('max')
4  t=0:0.1:10
5  a=0:0.1:10
6  subplot(2,3,1)
7  plot2d3(t,a)//Ramp function
8  title('Ramp function','fontsize',5)
9  xlabel('Time','fontsize',5)
10 ylabel('Amplitude','fontsize',5)
11 b=0:0.1:10
12 c=1
13 subplot(2,3,2)//Unit step function
14 plot(b,c)
15 title('Unit step function','fontsize',5)
16 xlabel('Time','fontsize',5)
17 ylabel('Amplitude','fontsize',5)
18 d=0:0.1:10
19 f=d+3//time advance by 3 sec
```

```
e=0:0.1:10
subplot(2,3,3)
plot2d3(f,e)//Delayed ramp function
title('Delayed ramp function','fontsize',5)
xlabel('Time','fontsize',5)
ylabel('Amplitude','fontsize',5)
g=0:0.1:10
j=g-3//delay by 3 sec
h=1
subplot(2,3,5)
plot(j,h)//Advanced unit step function
title('Advanced unit step function','fontsize',5)
xlabel('Time','fontsize',5)
ylabel('Amplitude','fontsize',5)
```

Experiment: 6

To study time shift and time advance

Scilab code Solution 6.6 time shifting

```
1  //Experiment-6
2  //Scilab 5.4.1, Windows 7
3  stacksize('max')
4  t=-5:0.01:5//time axis
5  k=gca()
6  x=%e^(-2.*t)//given signal
7  subplot(2,2,1.5)
8  plot(t,x)
9  k.thickness=2
10 k.y_location="middle"//axis location
11 title('original signal','fontsize',5)
12 xlabel('Time(in sec)','fontsize',5)
13 ylabel('Amplitude(in Volts)','fontsize',5)
14 a=%e^(-2.*t)
15 subplot(2,2,3)
16 k=gca()//get current axis in k
17 plot(t,a.*(t>1))//time delay by 1 sec
18 k.thickness=2
19 k.y_location="middle"
```

```
20 title('delayed original signal','fontsize',5)
21 xlabel('Time(in sec)','fontsize',5)
22 ylabel('Amplitude(in Volts)','fontsize',5)
23 b=%e^(-2.*t)
24 subplot(2,2,4)
25 k=gca()
26 plot(t,b.*(t>-1))//time advance by 1 sec
27 k.thickness=2
28 k.y_location="middle"
29 title('advanced original signal','fontsize',5)
30 xlabel('Time(in sec)','fontsize',5)
31 ylabel('Amplitude(in Volts)','fontsize',5)
```