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

Solutions provided by
Vivek Goklani
Electronics Engineering
VESIT

April 22, 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 Experiments

Experiment: 1

Convolution of discrete signal using convol and convolution sum

Scilab code Solution 1.1 convolution of discrete sequences

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

```
ylabel('h(n)','fontsize',5)
title('signal h(n)','fontsize',5)
subplot(2,3,3)
plot2d3(t2,y,rect=[0,0,10,100])
xlabel('t->','fontsize',5)
ylabel('y(n)','fontsize',5)
title('convolved signal (convol command)','fontsize'
    ,5)
printf("convolved signal using convol command\n")
disp(y)
printf("\n")

//verification using convolution sum formula
y=[0 0 0 0 0 0 0]
t=[1:7]
for k=1:4
    if((2-k)>0) then
     y(1)=y(1)+x(k)*h(2-k)
     end
end
for k=1:4
    if((3-k)>0) then
     y(2)=y(2)+x(k)*h(3-k)
     end
end
for k=1:4
    if((4-k)>0) then
     y(3)=y(3)+x(k)*h(4-k)
     end
end
for k=1:4
    if((5-k)>0) then
     y(4)=y(4)+x(k)*h(5-k)
     end
end
for k=1:4
    if((6-k)<5) then
     y(5)=y(5)+x(k)*h(6-k)
```

```
     end
end
for k=1:4
    if((7-k)<5) then
     y(6)=y(6)+x(k)*h(7-k)
     end
end
for k=1:4
    if((8-k)<5) then
     y(7)=y(7)+x(k)*h(7-k)
     end
end
y(7)=y(7)+4
printf("convolved sequence using convolution sum\n")
disp(y)
subplot(2,3,5)
plot2d3(t,y,rect=[0,0,10,100])
xlabel('t->','fontsize',5)
ylabel('y(n)','fontsize',5)
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

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

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

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

Experiment: 4

To find even and odd components of given signal

Scilab code Solution 4.4 even and odd signals

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

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

Experiment: 5

To implement representation of elementary signals using scilab

Scilab code Solution 5.5 elementary signals

```
//Experiment-5
//Scilab 5.4.1, Windows 7
stacksize('max')
t=0:0.1:10
a=0:0.1:10
subplot(2,3,1)
plot2d3(t,a)//Ramp function
title('Ramp function','fontsize',5)
xlabel('Time','fontsize',5)
ylabel('Amplitude','fontsize',5)
b=0:0.1:10
c=1
subplot(2,3,2)//Unit step function
plot(b,c)
title('Unit step function','fontsize',5)
xlabel('Time','fontsize',5)
ylabel('Amplitude','fontsize',5)
d=0:0.1:10
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

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

```
title('delayed original signal','fontsize',5)
xlabel('Time(in sec)','fontsize',5)
ylabel('Amplitude(in Volts)','fontsize',5)
b=%e^(-2.*t)
subplot(2,2,4)
k=gca()
plot(t,b.*(t>-1))//time advance by 1 sec
k.thickness=2
k.y_location="middle"
title('advanced original signal','fontsize',5)
xlabel('Time(in sec)','fontsize',5)
ylabel('Amplitude(in Volts)','fontsize',5)
```