Scilab Manual for
Signals & Systems
by Prof Akhtar Nadaf
Electronics and Telecommunication Engineering
N.K. Orchid College Of Engineering & Technology, Solapur[1]

Solutions provided by
Prof Akhtar Nadaf
Electronics and Telecommunication Engineering
N.k. Orchid College Of Engineering & Technology Solapur

October 17, 2021

[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

List of Figures

Experiment: 1

Generation of continuous & discrete time signals

Scilab code Solution 1.1 Generation of Continious Time Signals

```
// VERSION  : Scilab:6.0.1
// OS  :  WINDOWS  10
// CAPTION:GENERATION  OF  CONTINUOUS  TIME  SIGNALS
//UNIT  IMPULSE  SIGNAL
clc ;
clear all;
close ;
N =7; //SET  LIMIT
t1 = -N:N;
x1=[zeros(1,N),1,zeros(1,N)];
subplot (2,3,1);
plot (t1,x1);
xgrid(4,1,7); //xgrid([color],[thickness],[style])
xlabel("Time");
ylabel("Amplitude");
title("Unit  Impulse  Signal");

//UNIT  STEP  SIGNAL
t2 =0:4;
```

```
x2= ones (1,5);
subplot (2,3,2);
plot (t2,x2);
xgrid(4,1,7);
xlabel ("time");
ylabel ("amplitude");
title ("Unit Step Signal");

//EXPONENTIAL SIGNAL
t3 =0:1:20;
x3=exp(-t3);
subplot (2,3,3);
plot (t3,x3);
xgrid(4,1,7);
xlabel ("time");
ylabel ("Amplitude");
title ("Exponential Signal");

//RAMP SIGNAL
t4 = -0:20;
x4=t4;
subplot (2,3,4);
plot (t4 ,x4);
xgrid(4,1,7);
xlabel ("Time");
ylabel ( "Amplitude");
title ("Ramp Signal");

//SINUSOIDAL SIGNAL

t5 =0:0.04:1;
x5=sin(2*%pi*t5);
subplot(2,3,5);
plot (t5,x5);
xgrid(4,1,7);
title ("Sinusoidal SIgnal")
xlabel ("Time");
ylabel ("Amplitude");
```

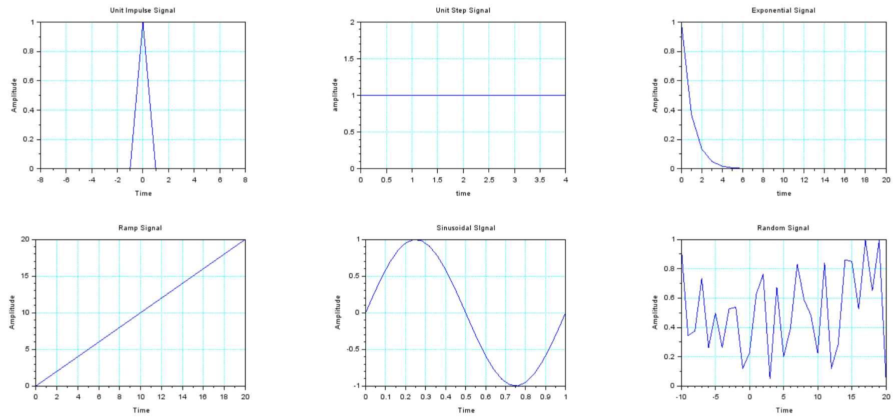

Figure 1.1: Generation of Continious Time Signals

```

//RANDOM SIGNAL
t6=-10:1:20;
x6=rand(1,31);
subplot(2,3,6);
plot(t6,x6);
xgrid(4,1,7);
xlabel("Time");
ylabel ("Amplitude");
title ("Random Signal");
```

Scilab code Solution 1.2 Generation of Discrete Time signals

```
//O.S. Windows 10
//Scilab 6.0.1
//Caption: Generation of Discrete time signals
clear;
clc;
// DT Unit step Signal
```

```
t =0:4;
y=ones(1,5);
subplot(3,2,1);
plot2d3(t,y);
xgrid(4,1,7); //xgrid([color],[thickness],[style])
xlabel("Time");
xlabel("n");
ylabel("Amplitude");
title("DT Unit Step  Signal");

//DT Unit Ramp signal
n1 =0:8;
//Figure 1.1: Waveform generation using DT signals
y1=n1;
subplot(3,2,2);
plot2d3(n1,y1);
xgrid(4,1,7); //xgrid([color],[thickness],[style])
xlabel("Time");
xlabel("n");
ylabel("Amplitude");
title ("DT Unit Ramp");

//DT Growing Exponential Signal
n1 =0:8;
y1=n1;
y2=exp(n1);
subplot (3 ,2 ,3);
plot2d3 (n1 ,y2);
xgrid(4,1,7); //xgrid([color],[thickness],[style])
xlabel("Time");
xlabel ("n");
ylabel ("Amplitude");
title("DT Growing Exponential signal");

//DT Decaying exponential Signal
n1 =0:8;
y1=n1;
y2=exp(-n1);
```

```
subplot (3 ,2 ,4);
plot2d3 (n1 ,y2);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("DT Decaying Exponential Signal" );

//DT Sinusoidal signal
n1 =0:25;
y1=n1;
y2=sin(n1);
subplot (3 ,2 ,5);
plot2d3 (n1 ,y2);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude" );
title("DT Sinusoidal Signal");

//DT Unit Impulse signal
l=7;
n=-l:l;
x=[zeros(1,l),1,zeros(1,l)];
b=gca();
b.y_location ="middle";
subplot(3,2,6);
plot2d3( "gnn",n,x);
xgrid(4,1,7);
a= gce ();
a.children (1).thickness =5;
xtitle("DT Unit sample sequence");
xlabel("n");
ylabel("Amplitude" );
```

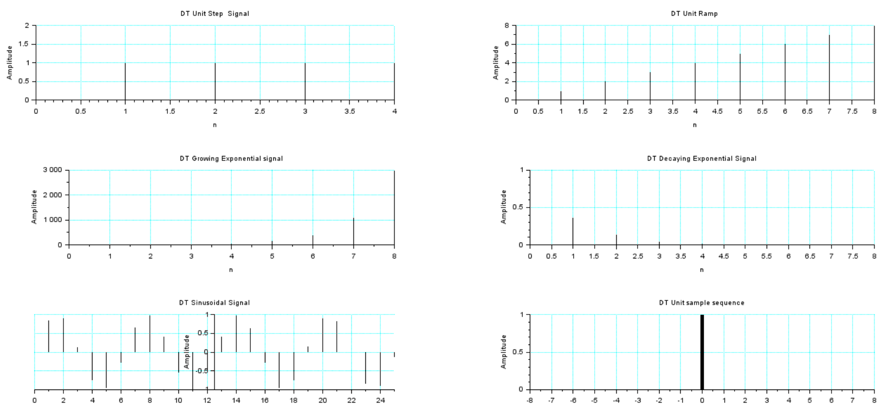

Figure 1.2: Generation of Discrete Time signals

Experiment: 2

Basic operations on CT & DT Signals

Scilab code Solution 2.1 Basic Operation on CT Signals

```
// VERSION :Scilab:6.0.1
// OS : WINDOWS 10
// CAPTION:BASIC OPERATION ON CONTINUOUS TIME
   SIGNALS
clc ;
clear all;
close ;
//Input CT Signal
t=-2:1:2;
x=[1,1,0,1,0];
subplot (2,3,1);
plot(t,x);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("Input CT Signal x(t)");

//Amplitude Scaling
a=2; // Scaling factor
```

```
subplot (2,3,2);
plot(t,a*x);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("z(t)=2x(t)");

//Amplitude Scaling
a=0.5; // Scaling factor
subplot (2,3,3);
plot(t,a*x);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("z(t)=0.5x(t)");

// Time reversal
subplot (2,3,4);
plot(-t,x);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("z(t)=x(-t)");

//Time Shifting
subplot (2,3,5);
plot(t+2,x);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("z(t)=x(t+2)");

//Time Shifting
subplot (2,3,6);
plot(t-2,x);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
```

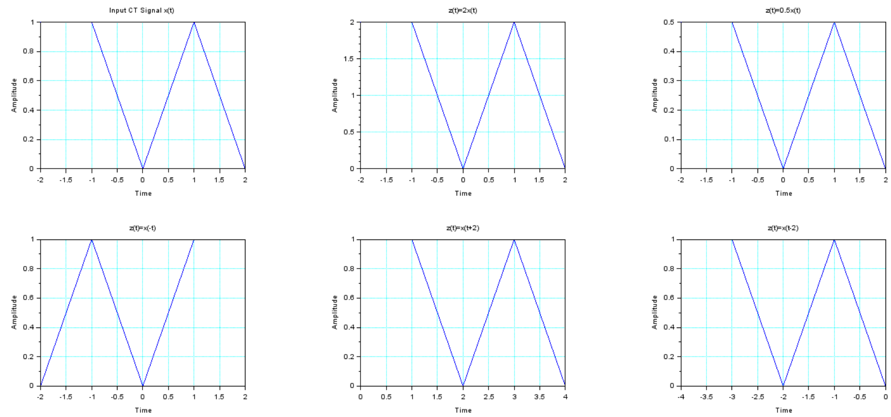

Figure 2.1: Basic Operation on CT Signals

```
title("z(t)=x(t-2)");
```

Scilab code Solution 2.2 Basic Operation on CT Signals

```
// VERSION :Scilab:6.0.1
// OS : WINDOWS 10
// CAPTION:BASIC OPERATION ON CONTINUOUS TIME SIGNALS
clc ;
clear all;
close ;
//Input CT Signal
t=-2:1:2;
x=[1,1,0,1,0];
subplot (2,3,1);
plot(t,x);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
```

```
title("Input CT Signal x(t)");

//Time Scaling
a=2; // Scaling factor
subplot (2,3,2);
plot(a*t,x);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("y(t)=x(2*t)");

//Time Scaling
a=0.5; // Scaling factor
subplot (2,3,3);
plot(a*t,x);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("y(t)=x(0.5*t)");

//Signal Addition
//Input Signal y(t)
t2=-2:1:2;
y=[1,1,0,1,1];
subplot (2,3,4);
plot(t2,y);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("Input signal y(t)");

//Signal addition
subplot (2,3,5);
plot(n,x+y);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("z(t)= x(t)+y(t)");
```

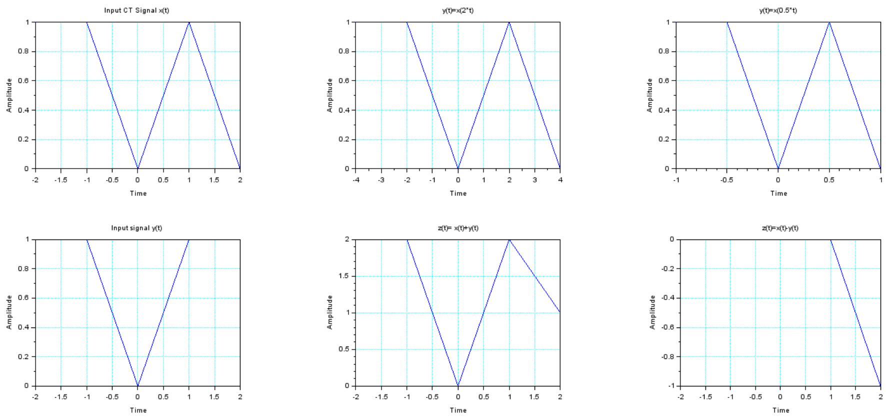

Figure 2.2: Basic Operation on CT Signals

```

//Signal subtraction
subplot (2,3,6);
plot(n,x-y);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("z(t)=x(t)-y(t)");
```

Scilab code Solution 2.3 Basic Operatiion on DT Signals

```
// VERSION :Scilab:6.0.1
// OS : WINDOWS 10
// CAPTION:BASIC OPERATION ON DISCRETE TIME SIGNALS
clc ;
clear all;
close ;
//Input DT Signal
n=-2:1:2;
```

```
x=[1,1,0,0.5,0.5];
subplot (2,3,1);
plot2d3(n,x,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("Input DT Signal x(n)");

//Amplitude Scaling
a=2; // Scaling factor
subplot (2,3,2);
plot2d3(n,a*x,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("z(n)=2*x(n)");

//Amplitude Scaling
a=0.5; // Scaling factor
subplot (2,3,3);
plot2d3(n,a*x,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("z(n)=0.5*x(n)");

// Time reversal
subplot (2,3,4);
plot2d3(-n,x,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("z(n)=x(-n)");

//Time Shifting
subplot (2,3,5);
plot2d3(n+2,x,style=[color("blue")]);
xgrid(4,1,7);
```

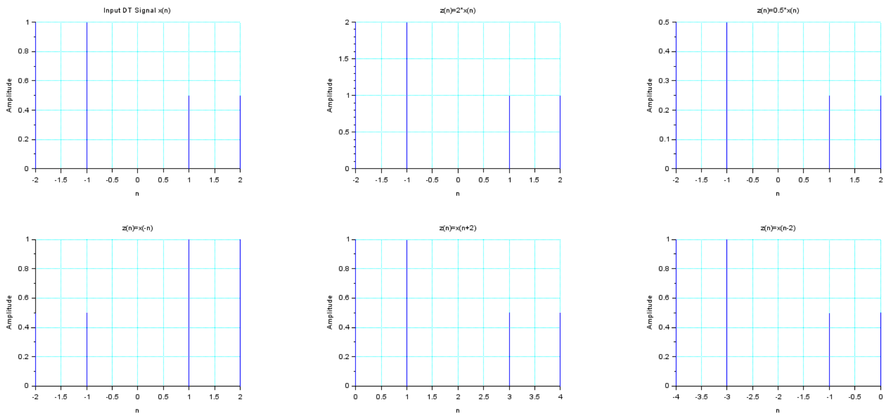

Figure 2.3: Basic Operatiion on DT Signals

```
xlabel("n");
ylabel("Amplitude");
title("z(n)=x(n+2)");

//Time Shifting
subplot (2,3,6);
plot2d3(n-2,x,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("z(n)=x(n-2)");
```

Scilab code Solution 2.4 Basic Operation on DT Signals

```
// VERSION :Scilab:6.0.1
// OS : WINDOWS 10
// CAPTION:BASIC OPERATION ON DISCRETE TIME SIGNALS
clc ;
clear all;
```

```
close ;
//Input DT Signal
n=-2:1:2;
x=[1,1,0,0.5,0.5];
subplot (2,3,1);
plot2d3(n,x,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("Input DT Signal x(n)");


//Time Scaling
a=2; // Scaling factor
subplot (2,3,2);
plot2d3(a*n,x,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("z(n)=2*x(n)");

//Time Scaling
a=0.5; // Scaling factor
subplot (2,3,3);
plot2d3(a*n,x,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("z(n)=0.5*x(n)");

//Signal Addition
//Input Signal y(n)
n2=-2:1:2;
y=[1,1,0,1,1];
subplot (2,3,4);
plot2d3(n2,y,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
```

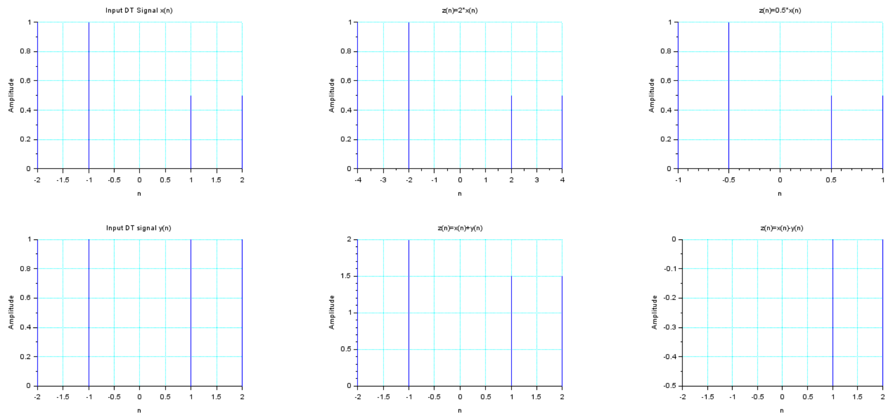

Figure 2.4: Basic Operation on DT Signals

```
ylabel("Amplitude");
title("Input DT signal y(n)");

//Signal addition
subplot (2,3,5);
plot2d3(n,x+y,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("z(n)=x(n)+y(n)");

//Signal subtraction
subplot (2,3,6);
plot2d3(n,x-y,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("z(n)=x(n)-y(n)");
```

Experiment: 3

Determination of Even & odd part of signal

Scilab code Solution 3.1 Determination of even and odd part of CT signal

```
// VERSION :Scilab:6.0.1
// OS : WINDOWS 10
// CAPTION:DETERMINATION OF EVEN & ODD PART OF CT
   SIGNAL
clc ;
clear all;
close ;
//Input CT Signal
t=-2:1:2;
x=[1,1,0,1,0];
subplot (2,3,1);
plot(t,x);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("Input CT Signal x(t)");
c=3;
for j=1:length(t)
```

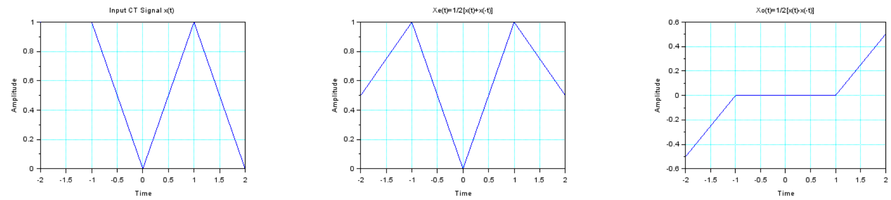

Figure 3.1: Determination of even and odd part of CT signal

```
    i=n(j);
    xe(j)=(1/2)*(x(c+i)+x(c-i));
    xo(j)=(1/2)*(x(c+i)-x(c-i));
end
xe=[xe(c-2),xe(c-1),xe(c+0),xe(c+1),xe(c+2)];
xo=[xo(c-2),xo(c-1),xo(c+0),xo(c+1),xo(c+2)];
subplot (2,3,2);
plot(t,xe);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("Xe(t)=1/2[x(t)+x(-t)]");
subplot (2,3,3);
plot(t,-xo);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("Xo(t)=1/2[x(t)-x(-t)]");
```

Scilab code Solution 3.2 Determination of even and odd part of DT signal

```
// VERSION : Scilab:6.0.1
// OS : WINDOWS 10
// CAPTION:DETERMINATION OF EVEN & ODD PART OF DT SIGNAL
clc ;
clear all;
close ;
//Input DT Signal
n=-2:1:2;
x=[-2,1,2,-1,3];
subplot (2,3,1);
plot2d3(n,x,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("Input DT Signal x(t)");
c=3;
for j=1:length(n)
    i=n(j);
    xe(j)=(1/2)*(x(c+i)+x(c-i));
    xo(j)=(1/2)*(x(c+i)-x(c-i));
end
xe=[xe(c-2),xe(c-1),xe(c+0),xe(c+1),xe(c+2)];//calculation of even Part
xo=[xo(c-2),xo(c-1),xo(c+0),xo(c+1),xo(c+2)];//ccalculation of odd part
subplot (2,3,2);
plot2d3(n,xe,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("Xe(n)=1/2[x(n)+x(-n)]");
subplot (2,3,3);
plot2d3(n,-xo,style=[color("blue")]);
xgrid(4,1,7);
```

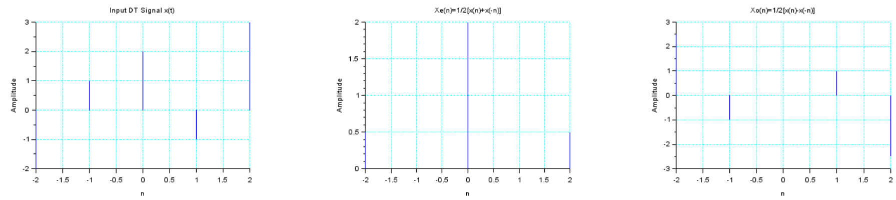

Figure 3.2: Determination of even and odd part of DT signal

```
xlabel("n");
ylabel("Amplitude");
title("Xo(n)=1/2[x(n)-x(-n)]");
```

Experiment: 4

Determination of energy & power of signal

Scilab code Solution 4.1 Determination of energy and power of CT signal

```
// VERSION :Scilab:6.0.1
// OS : WINDOWS 10
// CAPTION:DETERMINATION OF ENERGY & POWER OF SIGNAL
//Calculate energy of signal y(t)=5e^(-5t)
clc ;
clear all;
t =0:0.001:10;
y= 5*exp (-5*t);
E= integrate('(5*exp(-5*t))^2','t',0,2*%pi );// Calculation of energy of signal
disp (E,'Energy of given signal in Joules=' );// Expected output: 2.5 Joules
// Expected outpot in console
//Energy of given signal in Joules=
//    2.5
// Calculate power of x(t)=Ae^(-5t)
t=0:0.001:10;
y=5*exp (-5*t);
P= integrate('(5*exp(-5*t))^2','t',0,2*%pi )/(2*%pi)
```

```
;
disp(P,'power of the signal in Watts=:');
//Expected Output in console
// power of the signal in Watts=:
//    0.3978874
```

Experiment: 5

Sampling Theorem

Scilab code Solution 5.1 Sampling Theorem

```
// VERSION : Scilab:6.0.1
// OS : WINDOWS 10
// CAPTION: Sampling Theorem
clc ;
clear all;
t =0:0.01:100;
f =0.02;
x=sin(2*%pi*f*t);
subplot(2,2,1);
plot (t,x);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title ("CT Signal");
// Sampling frequency less than twices of input
   signal frequency
fs1 =0.002;
n =0:0.1:100;
x1=sin (2*%pi*f*n/fs1 );
subplot(2,2,2);
plot2d3(n,x1,style=[color("blue")]);
```

```
xlabel("n");
ylabel("Amplitude");
xgrid(4,1,7);
title( "fs<2f");
// Sampling frequency is equal to twices of input
   signal frequency
fs2 =0.04;
x2= sin (2*%pi*f*n/fs2);
subplot(2,2,3);
plot2d3(n,x2,style=[color("blue")]);
xlabel("n");
ylabel("Amplitude");
xgrid(4,1,7);
title ("Sampling fs=2f");
//Sampling frequency is greater than twice of input
   signal frequency
fs3 =0.4;
x3= sin (2*%pi*f*n/fs3 );
subplot (2,2,4);
plot2d3 (n,x3,style=[color("blue")]);
xlabel("n");
ylabel("Amplitude");
xgrid(4,1,7);
title ("fs>2f");
```

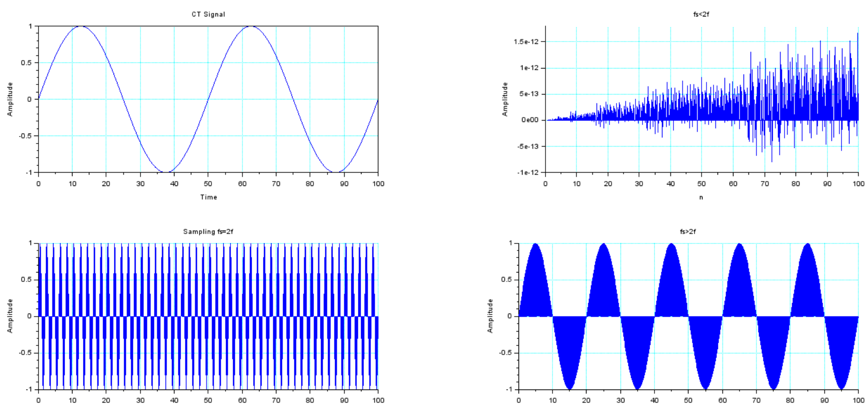

Figure 5.1: Sampling Theorem

Experiment: 6

Linear convolution sum and integral

Scilab code Solution 6.1 Linear convolution

```
// VERSION :Scilab:6.0.1
// OS : WINDOWS 10
// CAPTION: Convolution sum of two sequences x(n)
    =[1,4,3,2] & h(n)=[1,3,2,1]
clc ;
clear all;
clc;
x=[1 4 3 2];
t1=0:1:3
subplot(2,3,1);
plot2d3(t1,x,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("Input DT Signal x(n)");

h=[1 3 2 1];
t2=0:1:3
subplot(2,3,2);
```

```
plot2d3(t2,h,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("Input DT Signal h(n)");
y=convol(x,h);
disp(y,"Convolution sum of the above two sequences is:");
// Expected output in console:
//Convolution sum of the above two sequences is:

//    1.    7.    17.    20.    16.    7.    2.

l=length(y);
t3=0:l-1;
subplot(2,3,3);
plot2d3(t3,y,style=[color("blue")]);
xgrid(4,1,7);
xlabel("n");
ylabel("Amplitude");
title("y(n)=x(n)*h(n)");
```

Scilab code Solution 6.2 Convolution Integral

```
// Scilab Version: 6.0.1
//O.S : Windows 10
//Caption: Convolution integral of x(t)=(e^-at).u(t) and h(t)=u(t)
clear;
close;
clc;
T=10;
h=ones(1,T);
```

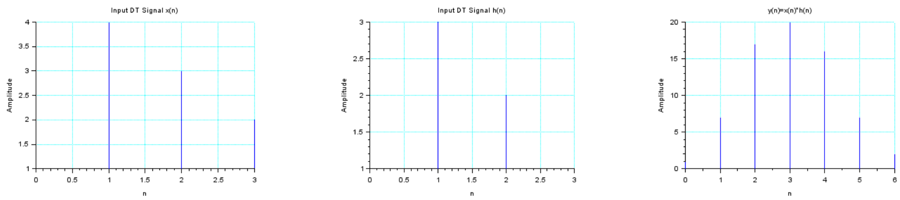

Figure 6.1: Linear convolution

```
N2=0:length(h)-1;
a=0.5;
for t=1:T
    x(t)=exp(-a*(t-1));
end
N1=0:length(x)-1;
y=conv(x,h)-1;
N=0:length(x)+length(h)-2;
subplot(2,3,1);
plot(N1,x)
xlabel("t");
ylabel("Amplitude");
xgrid(4,1,7);
title("Input signal x(t)=(e^-at).u(t)")
subplot(2,3,2);
plot(N2,h);
xlabel("t");
ylabel("Amplitude");
xgrid(4,1,7);
title("Input signal h(t)=u(t)");
subplot(2,3,3);
plot(N(1:T),y(1:T))
xlabel("t");
```

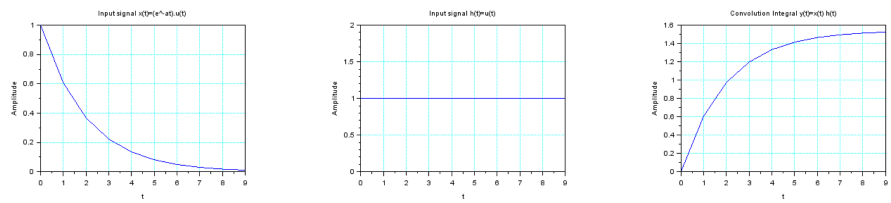

Figure 6.2: Convolution Integral

```
ylabel("Amplitude");
title("Convolution Integral y(t)=x(t) h(t)");
xgrid(4,1,7);
```

Experiment: 7

Fourier series

Scilab code Solution 7.1 Fourier Series of CT Signal

```
// Scilab Vrsion: 6.0.1
//O.S : Windows 10
//Caption: CTFS of a periodic signal x(t)
clear;
close;
clc;
t = -3:0.01:3;
//t1 = -%pi*4:(%pi*4)/100:%pi*4;
//t2 =-%pi*6:(%pi*6)/100:%pi*6;
xot = ones(1,length(t));
x1t = (1/2)*cos(%pi*2*t);
xot_x1t = xot+x1t;
x2t = cos(%pi*4*t);
xot_x1t_x2t = xot+x1t+x2t;
x3t = (2/3)*cos(%pi*6*t);
xt = xot+x1t+x2t+x3t;
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
```

```
a.data_bounds=[-4,0;2 4];
plot(t,xot)
ylabel('t')
title('xot =1')
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,-3;2 4];
plot(t,x1t)
ylabel('t')
title('x1(t) =1/2*cos(2*pi*t)')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,0;2 4];
plot(t,xot_x1t)
ylabel('t')
title('xo(t)+x1(t)')
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,-2;4 2];
plot(t,x2t)
ylabel('t')
title('x2(t) =cos(4*pi*t)')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,0;4 4];
plot(t,xot_x1t_x2t)
```

```
ylabel('t')
title('xo(t)+x1(t)+x2(t)')
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,-3;4 3];
plot(t,x3t)
ylabel('t')
title('x1(t) =2/3*cos(6*pi*t)')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,-3;4 3];
plot(t,xt)
ylabel('t')
title('x(t)=xo(t)+x1(t)+x2(t)+x3(t)')
```

Scilab code Solution 7.2 Fourier Series of DT Signal

```
// Scilab Version: 6.0.1
//O.S : Windows 10
//Caption: Discrete Time Fourier Transform of x[n]=
    sin(Won)
```

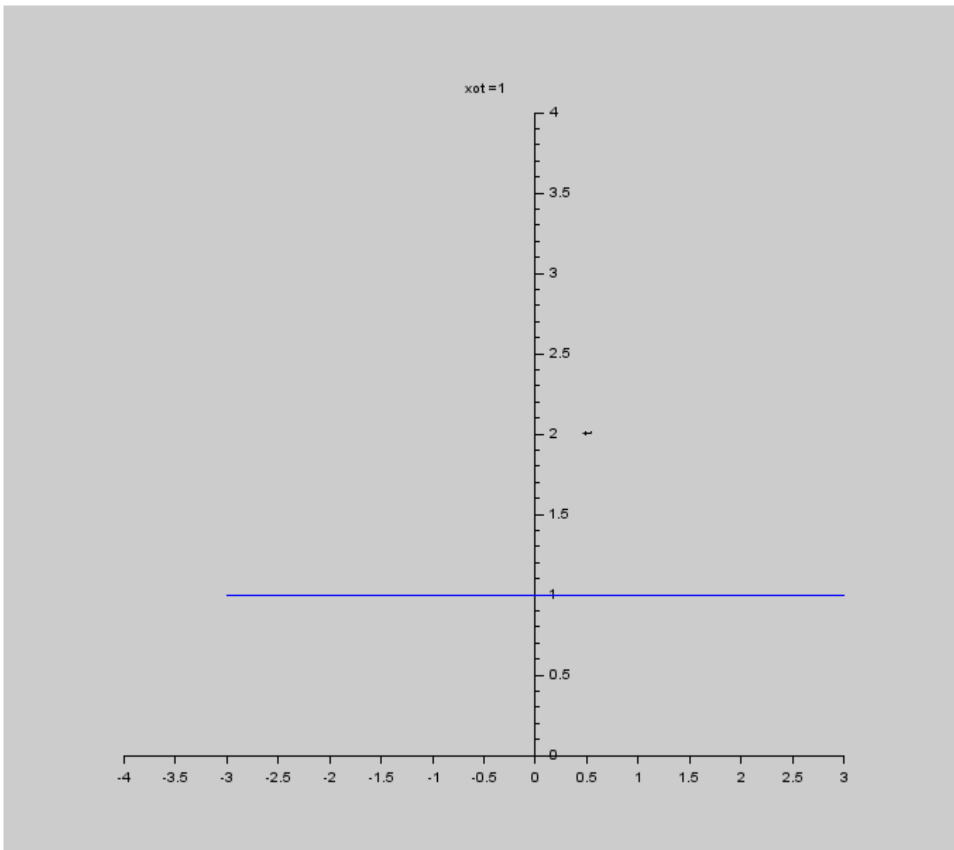

Figure 7.1: Fourier Series of CT Signal

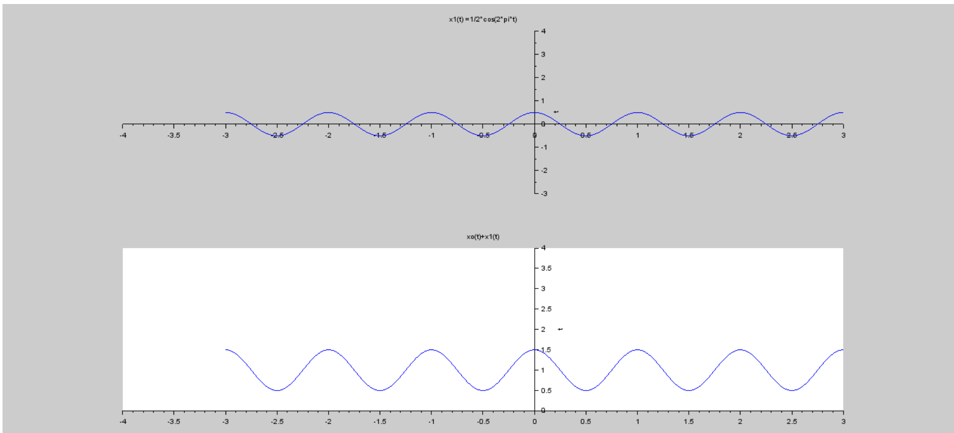

Figure 7.2: Fourier Series of CT Signal

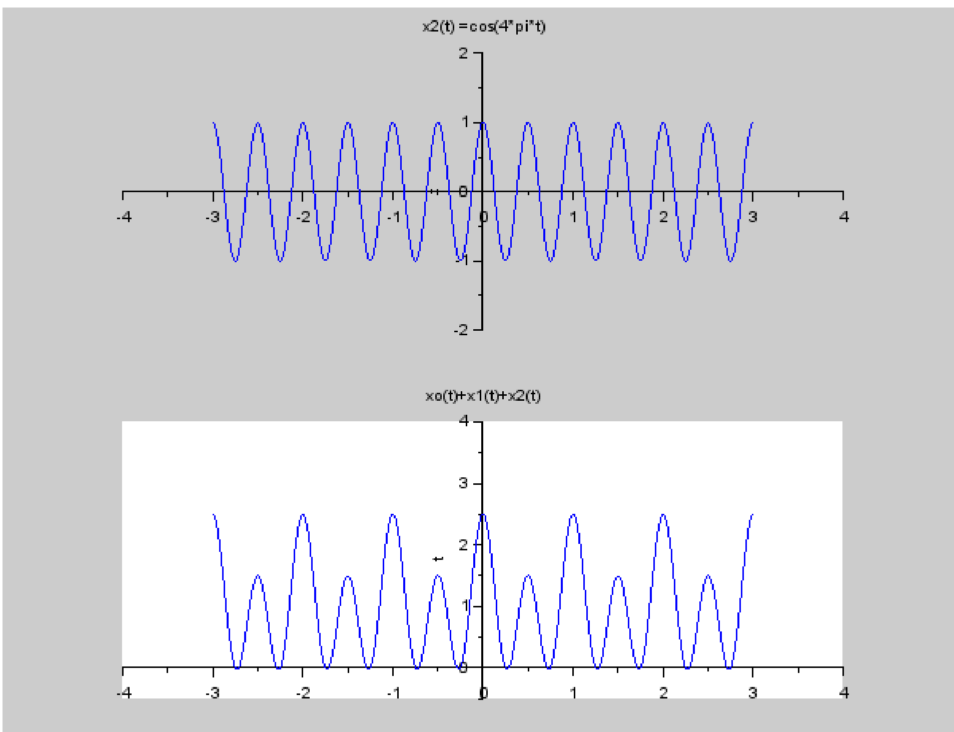

Figure 7.3: Fourier Series of CT Signal

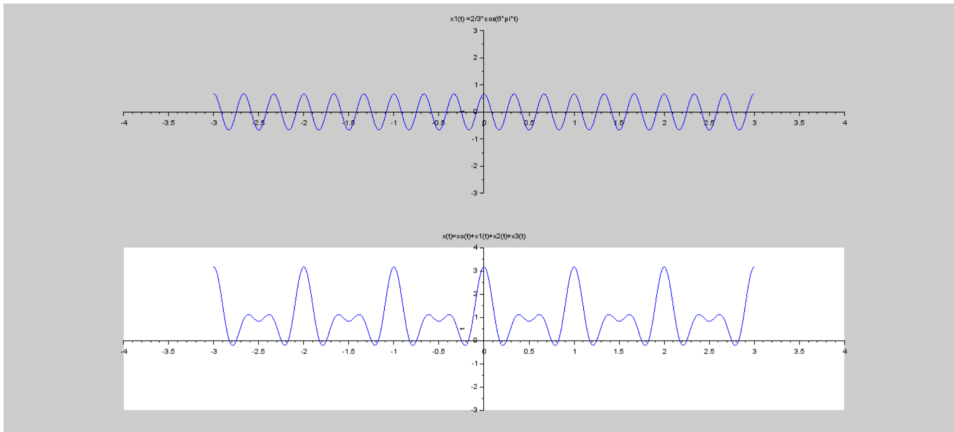

Figure 7.4: Fourier Series of CT Signal

```
clear;
close;
clc;
n = 0:0.01:5;
N = 5;
Wo = 2*%pi/N;
xn = sin(Wo*n);
for k =0:N-2
  C(k+1,:) = exp(-sqrt(-1)*Wo*n.*k);
  a(k+1) = xn*C(k+1,:)'/length(n);
  if(abs(a(k+1))<=0.01)
    a(k+1)=0;
  end
end
a =a'
a_conj = conj(a);
ak = [a_conj($:-1:1),a(2:$)]
k = -(N-2):(N-2);
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
```

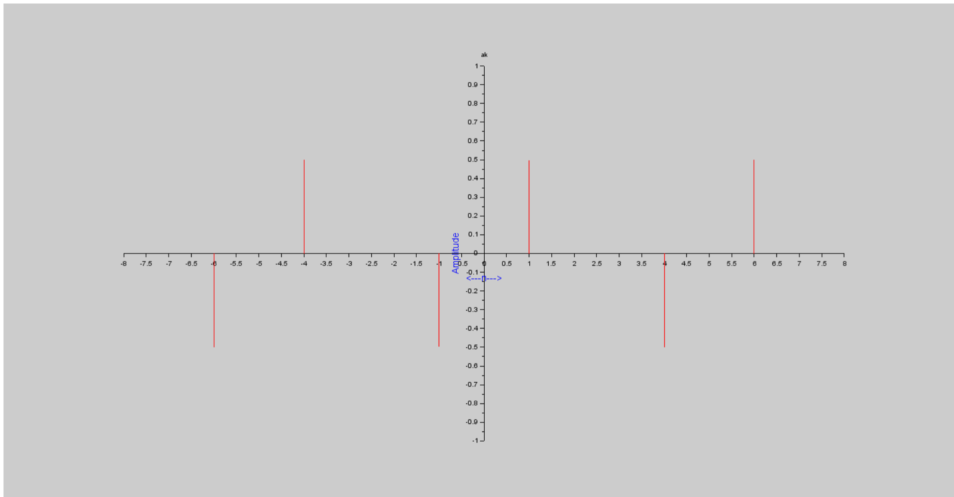

Figure 7.5: Fourier Series of DT Signal

```
a.data_bounds=[-8,-1;8,1];
poly1 = a;
poly1.thickness = 3;
plot2d3('gnn',k,-imag(ak),5)
xlabel("<——n——>","fontsize", 3, "color", "blue");
ylabel("Amplitude","fontsize", 3, "color", "blue");
poly1 = a;
poly1.thickness = 3;
plot2d3('gnn',N+k,-imag(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',-(N+k),-imag(ak($:-1:1)),5)
poly1 = a;
poly1.thickness = 3;
title('ak')
```

Experiment: 8

Fourier transform

Scilab code Solution 8.1 Fourier Transform of CT Signal

```
// Scilab Version: 6.0.1
//O.S : Windows 10
//Continuous Time Fourier Transform of h(t)=exp(-At)
   u(t), t>0
clear ;
clc ;
close ;
// Analog Signal
A =1; // Amplitude
Dt= 0.005;
t=0:Dt:10;
ht=exp(-A*t);
// Cont inuous time  Fourier Transform
Wmax=2*%pi*1; // Analog Frequency = 1Hz
K=4;
k=0:(K/1000):K;
W=k*Wmax/K;
HW=ht*exp(-sqrt(-1)*t'*W)*Dt;
HW_Mag=abs(HW);
W=[-mtlb_fliplr(W),W(2:1001)]; // Omega from Wmax to
    Wmax
```

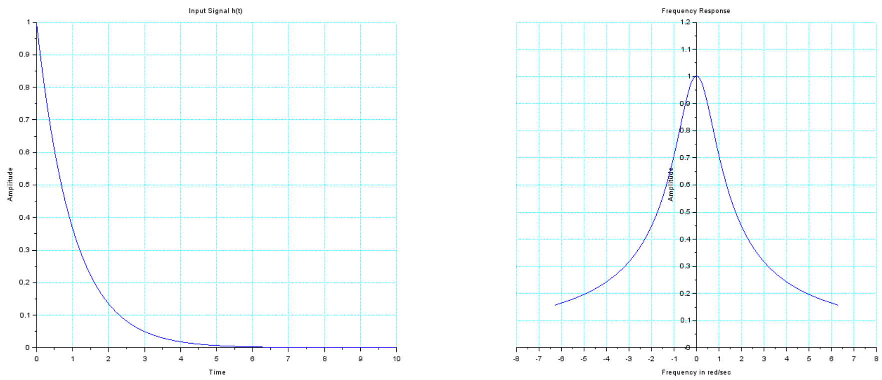

Figure 8.1: Fourier Transform of CT Signal

```
HW_Mag=[ mtlb_fliplr(HW_Mag),HW_Mag(2:1001)];
//Plotting Continuous Time Signal
subplot(1,2,1);
a=gca ();
a.y_location="origin";
plot(t,ht);
xgrid(4,1,7);
xlabel("Time");
ylabel("Amplitude");
title("Input Signal h(t)")
//Plotting Magnitude Responseof CTS
subplot(1,2,2);
a=gca();
a.y_location="origin";
plot(W,HW_Mag);
xgrid(4,1,7);
xlabel("Frequency in red/sec");
ylabel("Amplitude");
title("Frequency Response");
```

Scilab code Solution 8.2 Discrete Time Fourier Transform

```
//O.S : Windows 10
//Scilab Version: 6.0.1
//Caption: Discrete Time Fourier Transform of x[n]=1 , abs(n)<=N1
clear;
close;
clc;
// DTS Signal
N1 = 2;
n=-N1:N1;
x=ones(1,length(n));
// Discrete-time Fourier Transform
Wmax=2*%pi;
K =4;
k =0:(K/1000):K;
W =k*Wmax/K;
XW =x*exp(-sqrt(-1)*n'*W);
XW_Mag=real(XW);
W=[-mtlb_fliplr(W), W(2:1001)]; // Omega from -Wmax to Wmax
XW_Mag=[mtlb_fliplr(XW_Mag),XW_Mag(2:1001)];
//plot for abs(a)<1
subplot(2,1,1);
xgrid(4,1,7);
xlabel("<------n------>","fontsize",2,"color", "blue");
ylabel("Amplitude","fontsize",2,"color","blue");
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',n,x);
title("Discrete Time Sequence x[n]","fontsize",2,"
```

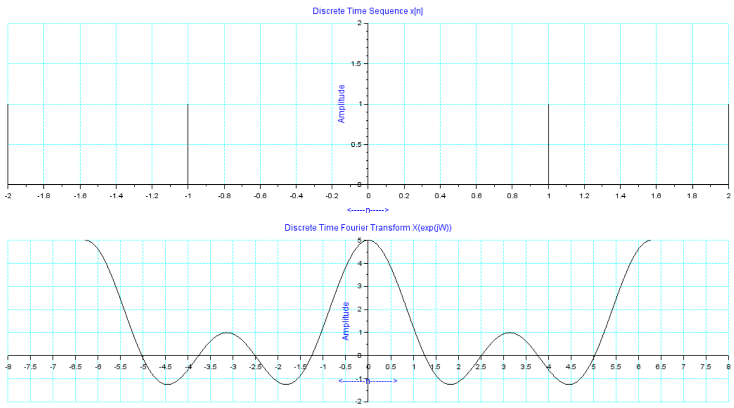

Figure 8.2: Discrete Time Fourier Transform

```
    color","blue");
subplot(2,1,2);
xgrid(4,1,7);
xlabel("<--------n-------->","fontsize",2,"color","
    blue");
ylabel("Amplitude", "fontsize",2, "color", "blue");
//ylabel("Amplitude");
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW_Mag);
title("Discrete Time Fourier Transform X(exp(jW))","
    fontsize",2,"color", "blue")'
```

Experiment: 9

Laplace Transform

Scilab code Solution 9.1 Laplace transform

```
// Scilab Version: 6.0.1
//O.S.: Windows 10
//Caption: Laplace transform of x(t)=exp(-0.4t).cos
   (12t)
clear;
clear all;
clc;
t=0:0.5:20;
ft=exp(-0.4*t).*sin(12*t);
s=%s;
numfs=s+0.4;
denfs=(s+0.4)^2+12^2;
fs=syslin('c',numfs/denfs);
fs1=csim('impulse',t,fs);
disp(fs,"Laplace Transform:-");
//disp(fs1,"Laplace Transform:-");

//Output:
//Laplace Transfprm:-


```

```
//              0.4 + s
//      -------------------------
//                                  2
//      144.16 + 0.8s + s
```

Experiment: 10

Z transform & Inverse Z transform

Scilab code Solution 10.1 Z transform and Inverse Z transform of given sequence

```
//Expt2: To Z- transform of [1 0 3 -1 2]
//O.S: Windows 10;
//Scilab: 6.0.1
clear;
clc ;
clear;
close ;
function[za]=ztransfer(sequence,n)
z=poly(0,'z','r')
za=sequence*(1/z)^n'
endfunction
x1=[1 0 3 -1 2];
n=0:length(x1)-1;
zz=ztransfer(x1,n);
//Display the result in command window
disp (zz,"Z-transform of sequence is:");
// Expected Output:
//Z-transform of sequence is:
```

```
//                2    4
//   2 − z + 3z + z
//  ──────────────────
//            4
//           z
disp('ROC is the entire plane except z = 0');
//ROC is the entire plane except z = 0

//Inverse  Z transform
z=%z;
a =(2+2* z+z ^2) ;
b=z^2;
h=ldiv(b,a,6);
disp(h," The Inverse Z    Transform i s ");

// The Inverse  Z Transform  is
// 1 .
//     2 .
// 2 .
// 0 .
//     4 .
// 8 .
```

Experiment: 11

Response of LTI System

This code can be downloaded from the website wwww.scilab.in

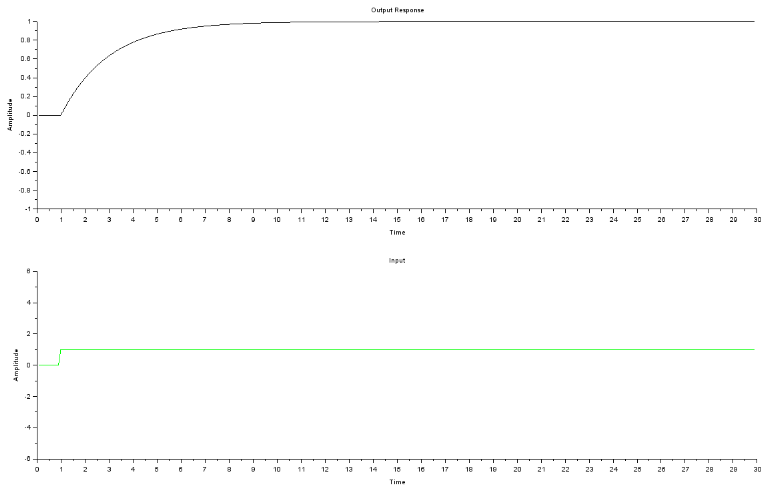

Figure 11.1: Step Response of LTI System

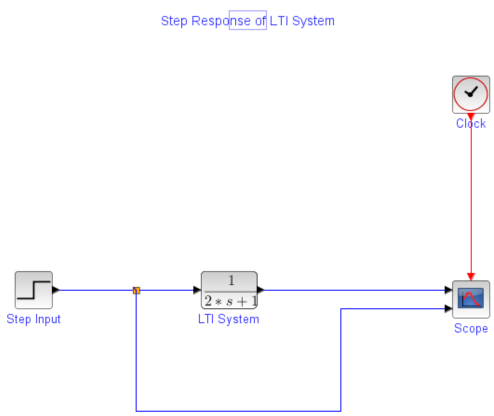

Figure 11.2: Step Response of LTI System

Experiment: 12

Probability

Scilab code Solution 12.1 Estimation of probability for tossing of coin

```
//Experiment no: 12
//Scilab Version: 6.0.1
//O.S.: Windows 10
//Program to estimate probability of tossing of a
   coin
clc;
close;
clear all;
N=input("Number of times coin is to be tossed:")
// Input: Number of times coin is to be tossed:1
NH=0; // Counter for head
NT=0; //counter for tail

for i=1:N
    p=grand(1,"prm",(1:2)')';

    if p(1) == 1;
        NT=NT+1;
    else
        NH=NH+1;
        end
```

```
end
PH=NH/N;
PT=NT/N
disp(PH, "Probability of outcome head :");
disp(PT,"Probability of outcome tail :");

// Output:
//Probability of outcome head :

//    1.

// Probability of outcome tail :

//    0.
```

Experiment: 13

Auto correlation

Scilab code Solution 13.1 Auto Correlation of DT signal

```
//Experiment no 13 Auto Correlation
// SciLab version : 6.0.1
// O.S. : Windows 10
clear;
clc;
close;
x = input('Enter the given discrete time sequence');
    // Enter a sequence x(n)={1,2,3,4}
l = length(x);
t=0:l-1;
subplot(1,2,1);
plot2d3 (t,x,style=[color("blue")]);
xgrid(4,1,7);
xlabel('n');
ylabel('x(n)');
title('Input sequence x(n)');
h = zeros(1,l);
for i = 1:l
  h(l-i+1) = x(i);
end
N = 2*l-1;
```

```
Rxx = zeros(1,N);
for i = l+1:N
   h(i) = 0;
end
for i = l+1:N
    x(i) = 0;
end
for n = 1:N
  for k = 1:N
    if(n >= k)
      Rxx(n) = Rxx(n)+x(n-k+1)*h(k);
    end
  end
end
disp(Rxx,'Auto Correlation Result is');
//Expected Result Rxx(n) ={4,11,20,30,20,11,4}
L=length(Rxx);
t=0:L-1;
subplot(1,2,2);
plot2d3 (t,Rxx,style=[color("blue")]);
xgrid(4,1,7);
xlabel('n');
ylabel('Rxx(n)');
title('Auto correlation of x(n)');
```

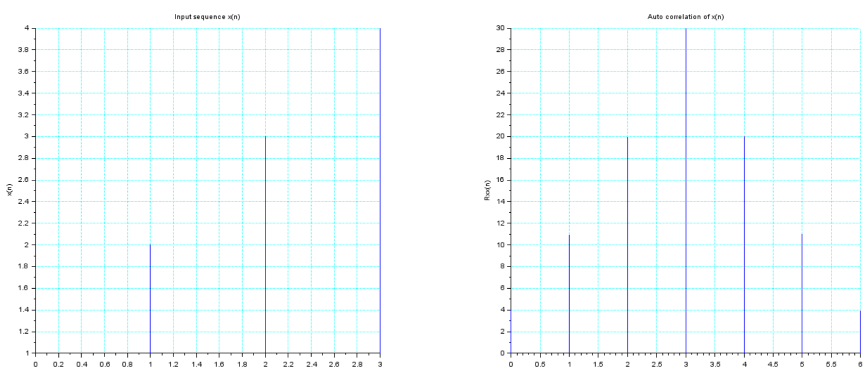

Figure 13.1: Auto Correlation of DT signal

Experiment: 14

Cross correlation

Scilab code Solution 14.1 Cross correlation

```
//Experiment no 14
//cross correlation
// SciLab version : 6.0.1
// O.S. : Windows 10
clc;
close ;
t1=0:4;
x1=[0,1,2,3,4];
subplot(2,2,1);
xgrid(4,1,7);
plot2d3 (t1,x1);
xlabel('n');
ylabel('x1(n)');
title('Input sequence x1(n)');

t2=0:4;
x2=[0,1,5,6,4];
subplot(2,2,2);
xgrid(4,1,7);
plot2d3 (t2,x2);
xlabel('n');
```

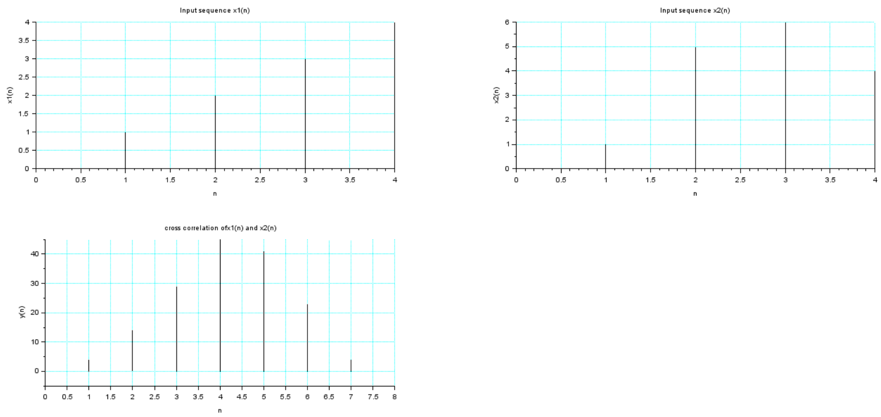

Figure 14.1: Cross correlation

```
ylabel('x2(n)');
title('Input sequence x2(n)');

y=xcorr(x1,x2);
l=length(y);
t3=0:l-1;
subplot(2,2,3);
plot2d3 (t3,y);
xgrid(4,1,7);
xlabel('n');
ylabel('y(n)');
title('cross correlation ofx1(n) and x2(n)');
```