Scilab Manual for
Digital Signal Processing
by Ms E. Sangeetha Devi
Electronics Engineering
Periyar Maniammai University[1]

Solutions provided by
Ms E.SANGEETHA DEVI
Electronics Engineering
PERIYAR MANIAMMAI UNIVERSITY,THANAJAVUR

August 5, 2025

[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 SIGNALS

Scilab code Solution 1.1 sinewave

```
clc;
clf;
clear all;
//Caption:generation of sine wave
f=0.2;
t=0:0.1:10;
x=sin(2*%pi*t*f);
plot(t,x);
title('sine wave');
xlabel('t');
ylabel('x');
```

Scilab code Solution 1.2 cosine wave

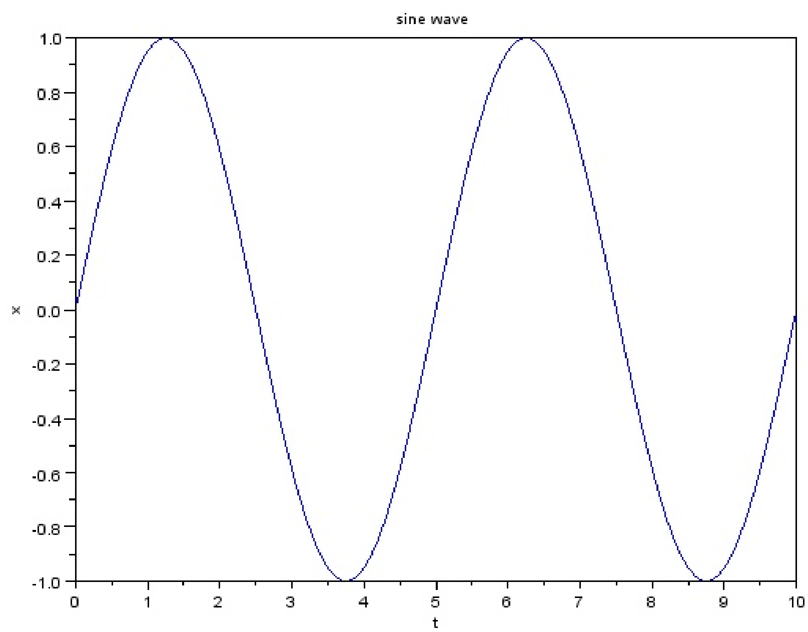

Figure 1.1: sinewave

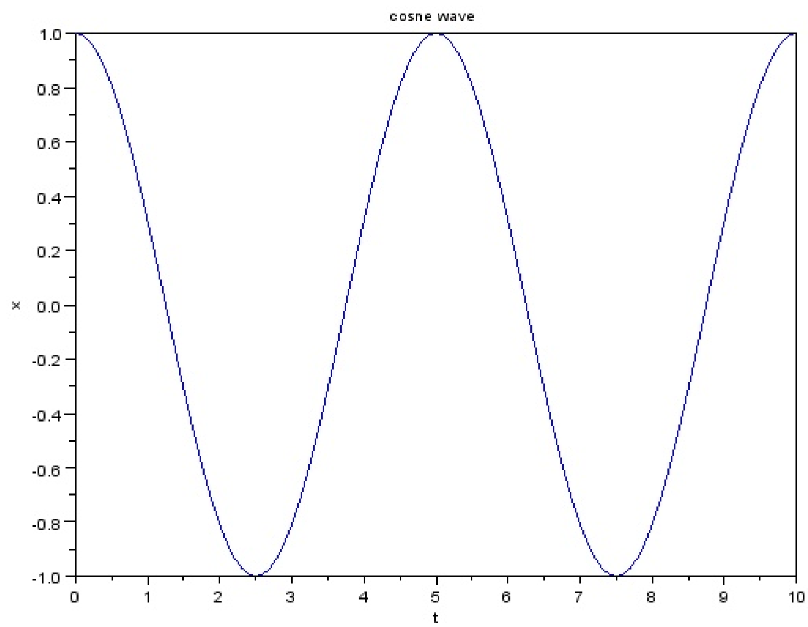

Figure 1.2: cosine wave

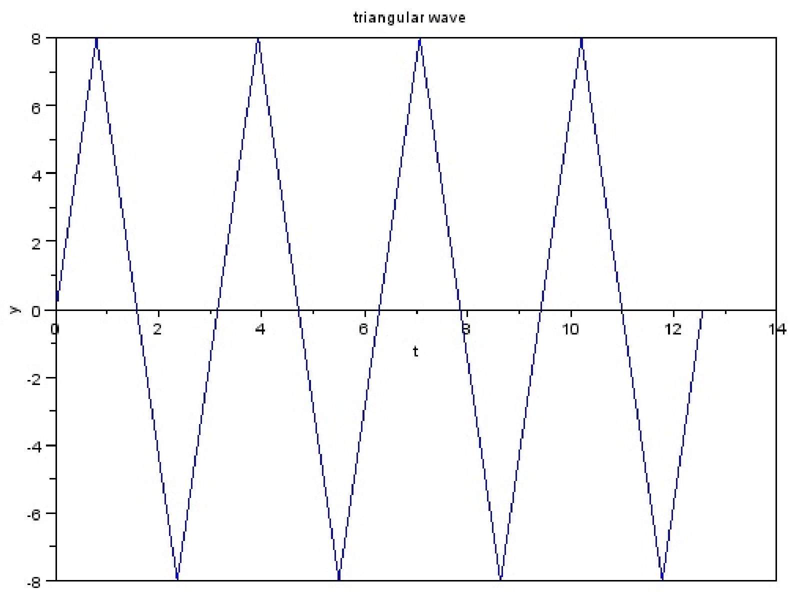

Figure 1.3: triangular wave

```
clc;
clf;
clear all;
//Caption:generation of cosine wave
f=0.2;
t=0:0.1:10;
x=cos(2*%pi*t*f);
plot(t,x);
title('cosne wave');
xlabel('t');
ylabel('x');
```

Scilab code Solution 1.3 triangular wave

```
clc;
clf;
clear all;
//Caption:generation of triangular wave
a=8;
t=0:(%pi/4):(4*%pi);
y=a*sin(2*t);
a=gca();
a.x_location="middle"
plot(t,y);
title('triangular wave');
xlabel('t');
ylabel('y');
```

Scilab code Solution 1.4 signum function

```
clc;
clf;
clear all;
//Caption:signum function
t =-5:0.1:5
a=gca();
a.x_location="middle"
x=sign(t);
b =gca();
b.y_location="middle"
plot(t,x);
title('signum function');
```

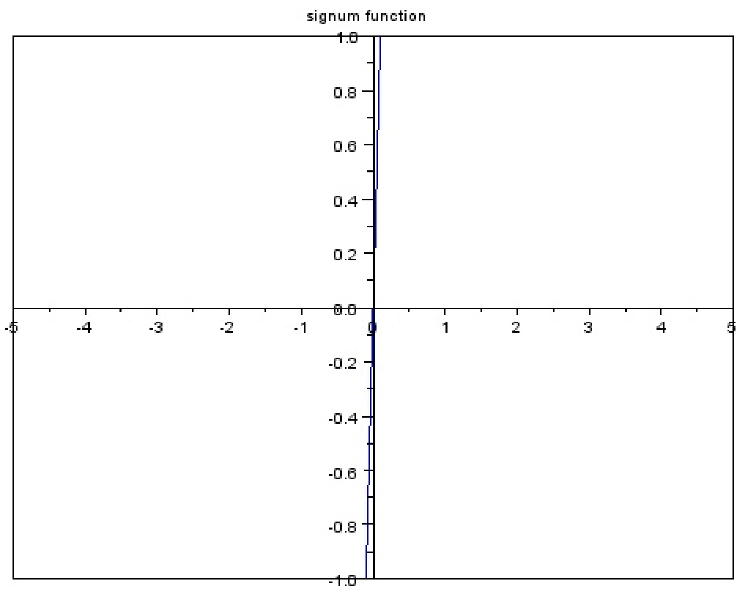

Figure 1.4: signum function

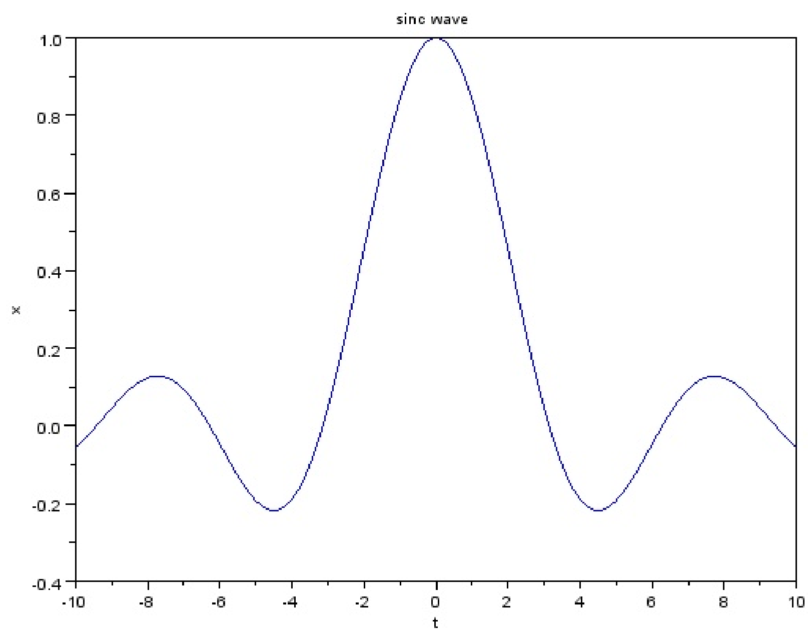

Figure 1.5: sinc function

Scilab code Solution 1.5 sinc function

```
clc;
clf;
clear all;
//Caption:sinc function
t=-10:0.1:10
x=sinc(t);
plot(t,x);
title('sinc wave');
xlabel('t');
ylabel('x');
```

Scilab code Solution 1.6 Exponential wave

```
clc;
clf;
clear all;
//Caption:generation of exponential wave
t=-2:0.1:2;
x=exp(t);
plot(t,x);
title('exponential wave');
xlabel('t');
ylabel('x');
```

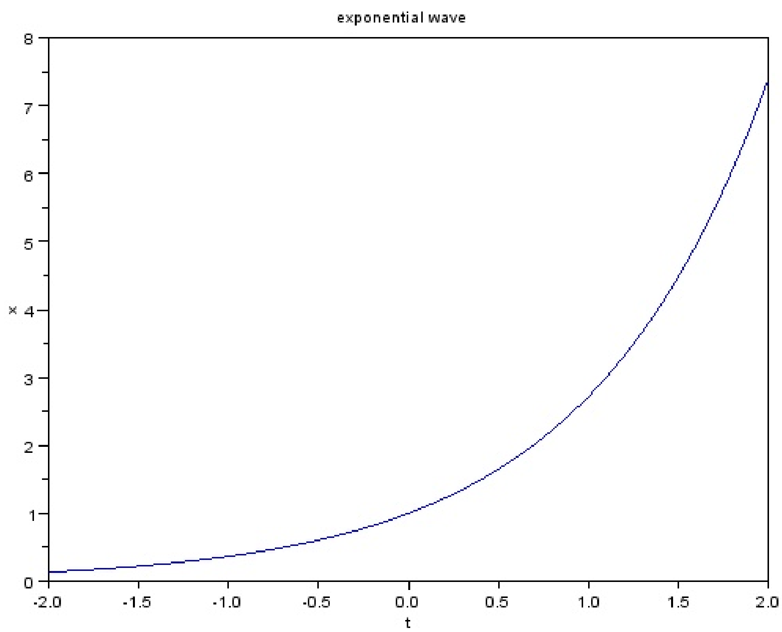

Figure 1.6: Exponential wave

Experiment: 2

GENERATION OF DISCRETE SIGNALS

Scilab code Solution 2.1 unit impulse signal

```
clc;
clf;
clear all;
//unit impulse
L=5;
n=-L:L;
x=[zeros(1,L),ones(1,1),zeros(1,L)];
a=gca();
a.y_location="middle"
plot2d3(n,x);
title('unit impulse');
```

Scilab code Solution 2.2 unitstepsignal

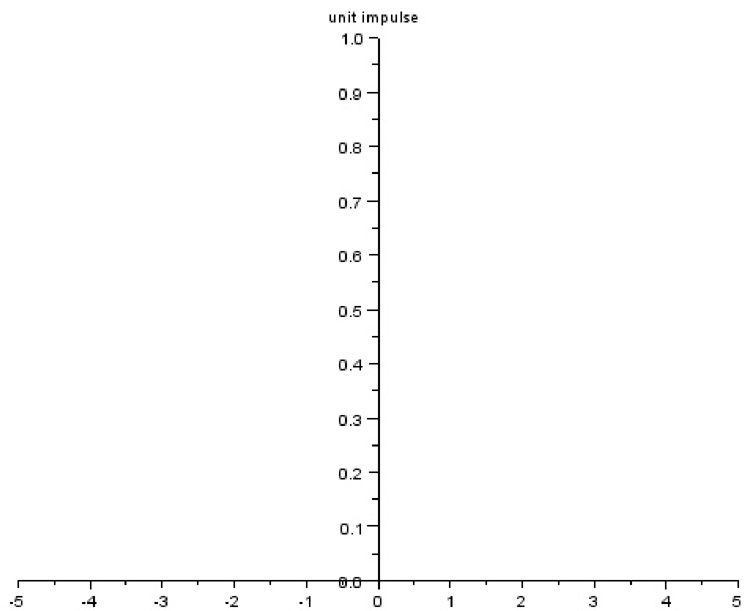

Figure 2.1: unit impulse signal

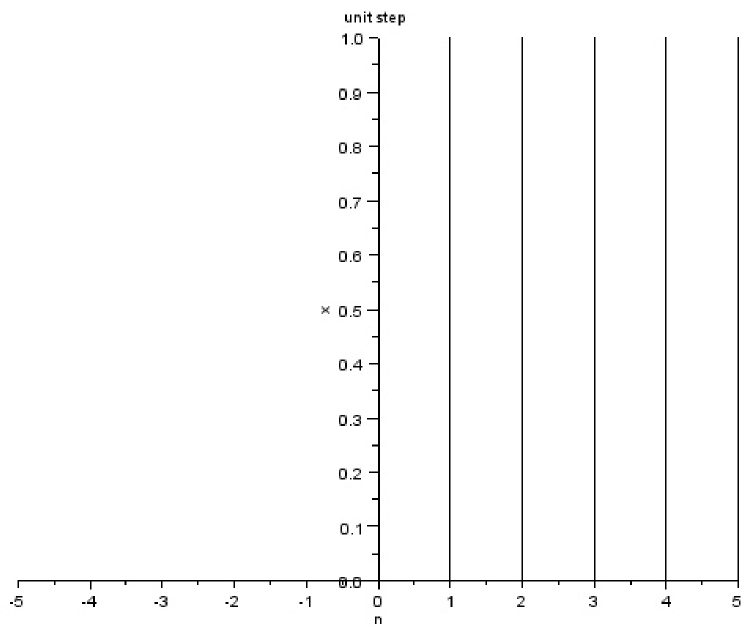

Figure 2.2: unitstepsignal

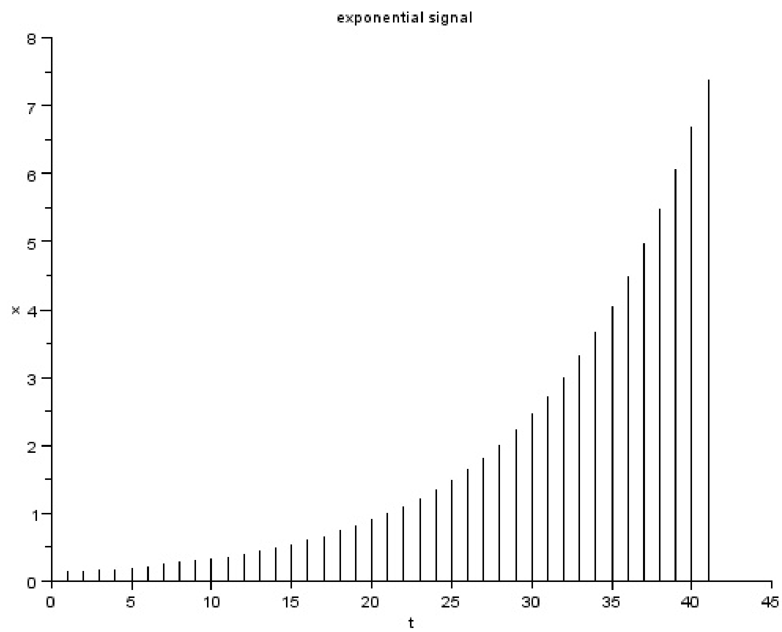

Figure 2.3: discreteexponentialwave

```
clc;
clf;
clear all;
L=5;
n=-L:L;
x=[zeros(1,L),ones(1,L+1)];
a=gca();
a.y_location="middle";
plot2d3(n,x);
title('unit step');
xlabel('n');
ylabel('x');
```

Scilab code Solution 2.3 discreteexponentialwave

```
//unit exponential
clc;
clf;
clear all;
a=1;
x=exp(a*t);
plot2d3(x);
title('exponential signal');
xlabel('t');
ylabel('x');
```

Scilab code Solution 2.4 unit ramp

```
//unit ramp
clc;
clf;
clear all;
L=5;
n=-L:L;
x=[zeros(1,L),0:L];
a =gca();
a.y_location='middle';
plot2d3(n,x);
xtitle('unit ramp signal');
xlabel('——>n');
ylabel('——>x(n)');
```

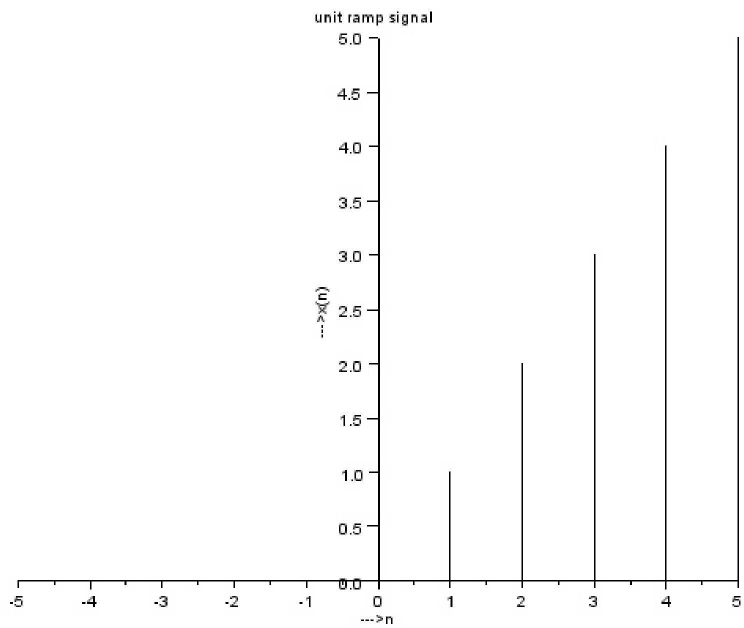

Figure 2.4: unit ramp

Experiment: 3

GENERATION OF SINUSOIDAL SIGNALS

Scilab code Solution 3.1 Generation of sinusoidal signals

```
clc;
clear all;
tic;
t=0:.01:%pi;
//generation of sine signals
y1=sin(t);
y2=sin(3*t)/3;
y3=sin(5*t)/5;
y4=sin(7*t)/7;
y5=sin(9*t)/9;
y = sin(t) + sin(3*t)/3 + sin(5*t)/5 + sin(7*t)/7 +
    sin(9*t)/9;
plot(t,y,t,y1,t,y2,t,y3,t,y4,t,y5);
legend('y','y1','y2','y3','y4','y5');
title('generation of sum of sinusoidal signals');
xgrid(1);
ylabel('——> Amplitude');
```

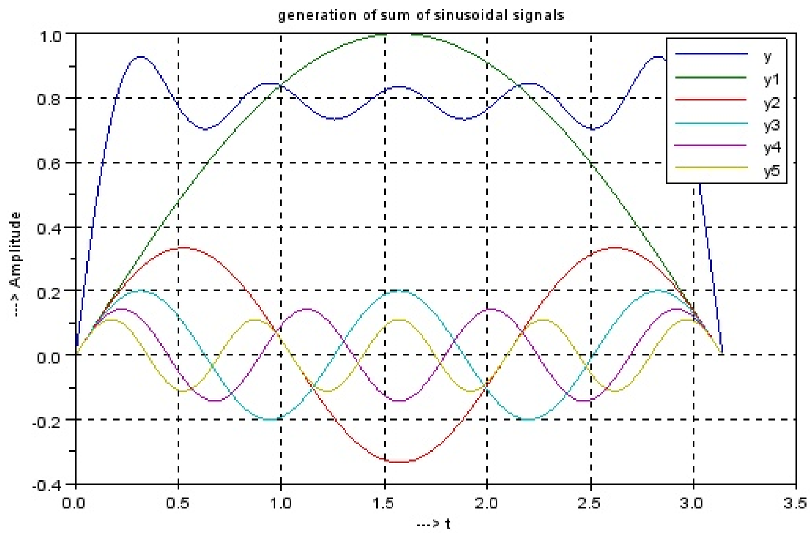

Figure 3.1: Generation of sinusoidal signals

```
xlabel('---> t');
toc;
```

Experiment: 4

GENERATION OF COMPLEX EXPONENTIAL SIGNALS

Scilab code Solution 4.1 generation of complex exponential signals

```
//program for generation of complex exponential
   signals
clc;
clf;
clear all;
n=-10:0.1:10;
a=0.8;
b=-0.8;
c=1.25;
d=-1.25;
x1=a^n;
subplot(2,2,1);
plot2d3(n,x1);
xlabel('n');
ylabel('amplitude');
```

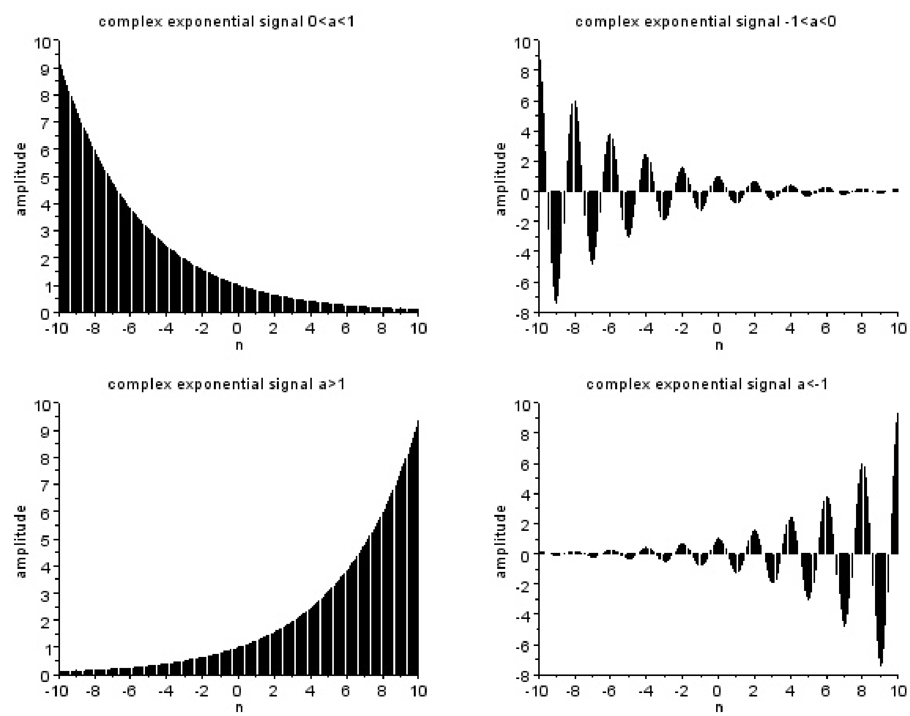

Figure 4.1: generation of complex exponential signals

```
title('complex exponential signal 0<a<1');
x2=b^n;
subplot(2,2,2);
plot2d3(n,x2);
xlabel('n');
ylabel('amplitude');
title('complex exponential signal -1<a<0');
x3=c^n;
subplot(2,2,3);
plot2d3(n,x3);
xlabel('n');
ylabel('amplitude');
title('complex exponential signal a>1');
x4=d^n;
subplot(2,2,4);
plot2d3(n,x4);
xlabel('n');
ylabel('amplitude');
title('complex exponential signal a<-1');
```

Experiment: 5

EXPONENTIALLY GROWING & DECAYING SIGNAL

Scilab code Solution 5.1 Exponentially Growing signals

```
//Exponentially decaying signal
clc;
clf;
clear all;
disp('Exponentially Decaying signal');
N=2;
a= 0.1;
n=0:0.1:N;
x=a.^-n;
disp('Exponentially growing signal');
disp(x);
plot2d3(n,x);
xlabel('Time');
ylabel('Amplitude');
title('Exponentially growing Signal Response');
```

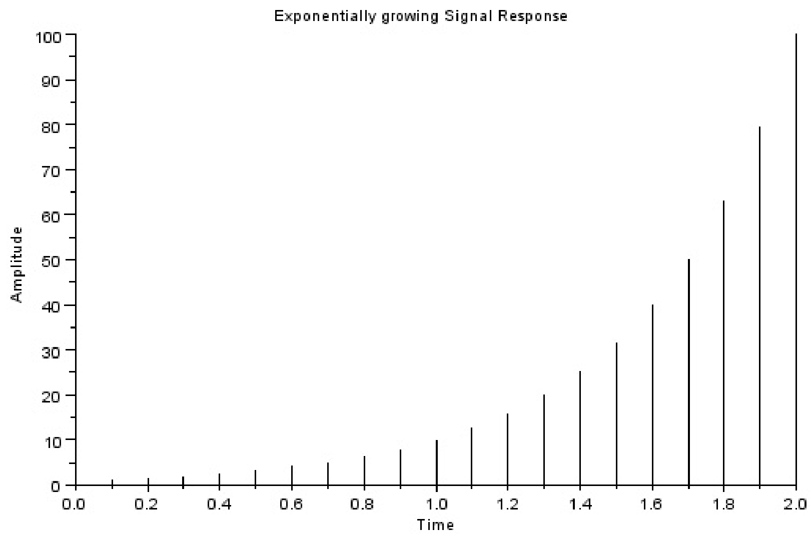

Figure 5.1: Exponentially Growing signals

Scilab code Solution 5.2 Exponentially decaying signals

```

clc;
clf;
clear all;
disp('Exponentially Decaying signal');
N=5;
a= 0.1;
n=0:0.1:N;
x=a.^n;
disp('Exponentially decaying signal');
disp(x);
```

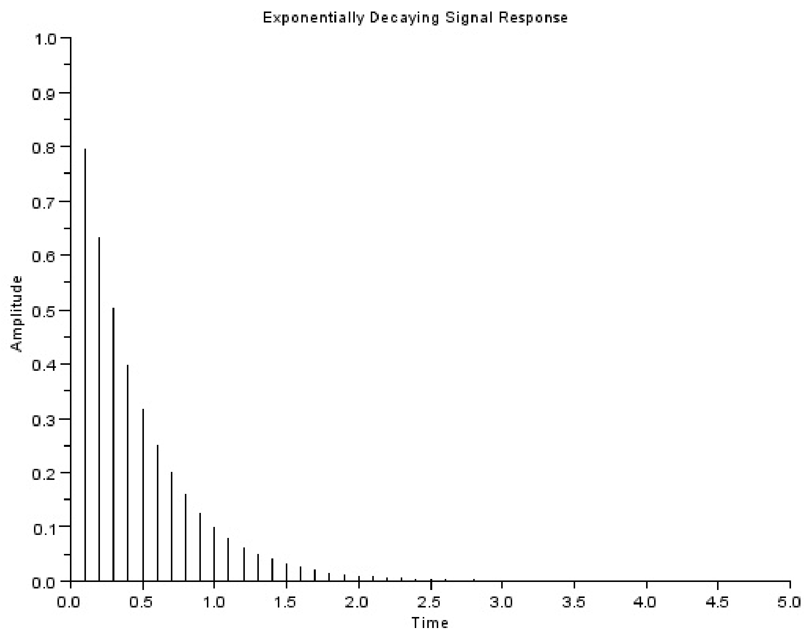

Figure 5.2: Exponentially decaying signals

```
plot2d3(n,x);
xlabel('Time');
ylabel('Amplitude');
title('Exponentially Decaying Signal Response');
```

Experiment: 6

ADDITION OF TWO CONTINUOUS SIGNALS

Scilab code Solution 6.1 Addition of two continuous signals

```
//program for addition of two continuous signals
clc;
clf;
clear all;
t=0:0.1:30
x1=sin(2*%pi*t/9);
x2=sin(2*%pi*t/2);
x3=x1+x2;
subplot(1,3,1);
plot(t,x1);
title('sinewave1')
xlabel('t');
ylabel('x1(t)');
subplot(1,3,2);
plot(t,x2);
title('sinewave2');
xlabel('t');
```

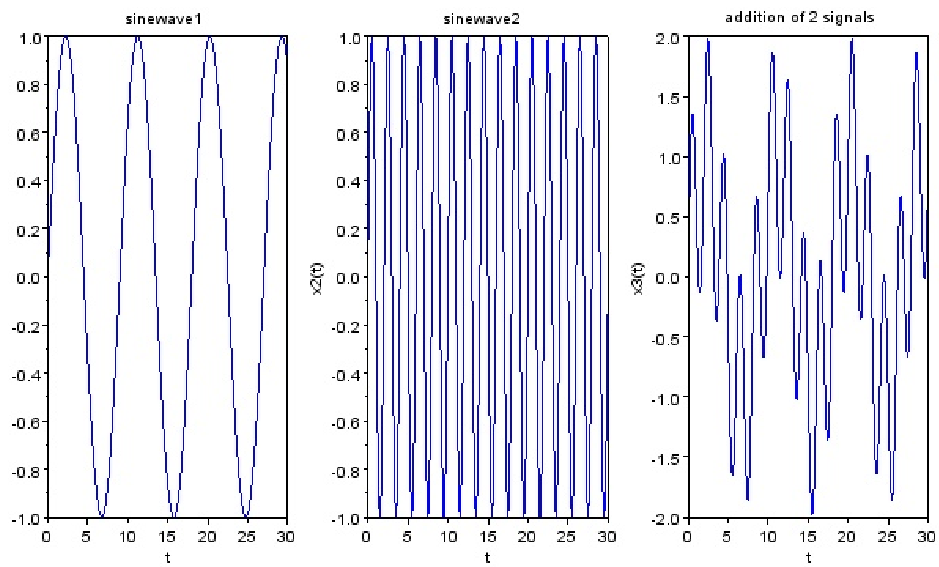

Figure 6.1: Addition of two continuous signals

```
ylabel('x2(t)');
subplot(1,3,3);
plot(t,x3);
title('addition of 2 signals');
xlabel('t');
ylabel('x3(t)');
```

Experiment: 7

ADDITION OF TWO DISCRETE SIGNALS

Scilab code Solution 7.1 Addition of two discrete signals

```
//program for addition of two discrete signals
N=5;
n=0:0.1:N-1;
x1=sin(1*%pi*n);
x2=sin(2*%pi*n);
x3=x1+x2;
figure(1);
subplot(1,3,1);
plot2d3(n,x1);
title('sine wave1')
xlabel('n');
ylabel('x1(n)');
subplot(1,3,2)
plot2d3(n,x2);
title('sinewave2')
xlabel('n');
ylabel('x2(n)');
```

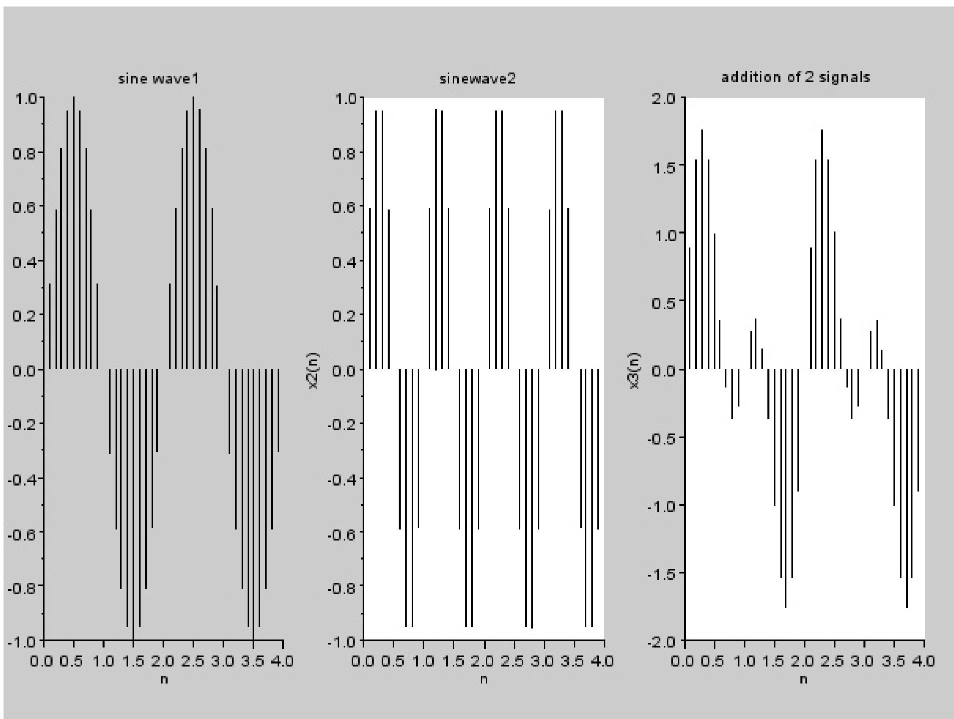

Figure 7.1: Addition of two discrete signals

```
subplot(1,3,3);
plot2d3(n,x3);
title('addition of 2 signals')
xlabel('n');
ylabel('x3(n)');
```

Experiment: 8

PROGRAM FOR SAMPLING PROCESS

Scilab code Solution 8.1 sampling

```
//program for sampling
clc;
clf;
clear all;
t=0:0.01:100;
fm=0.02;
x=cos(2*%pi*fm*t);
subplot(2,2,1);
plot(t,x);
title('continious time signal');
fs1=0.002;
n=0:1:50;
x1=cos(2*%pi*fm*n/fs1);
subplot(2,2,2);
plot2d3(n,x1);
title('time signal fs1<2fm');
fs2=0.04;
```

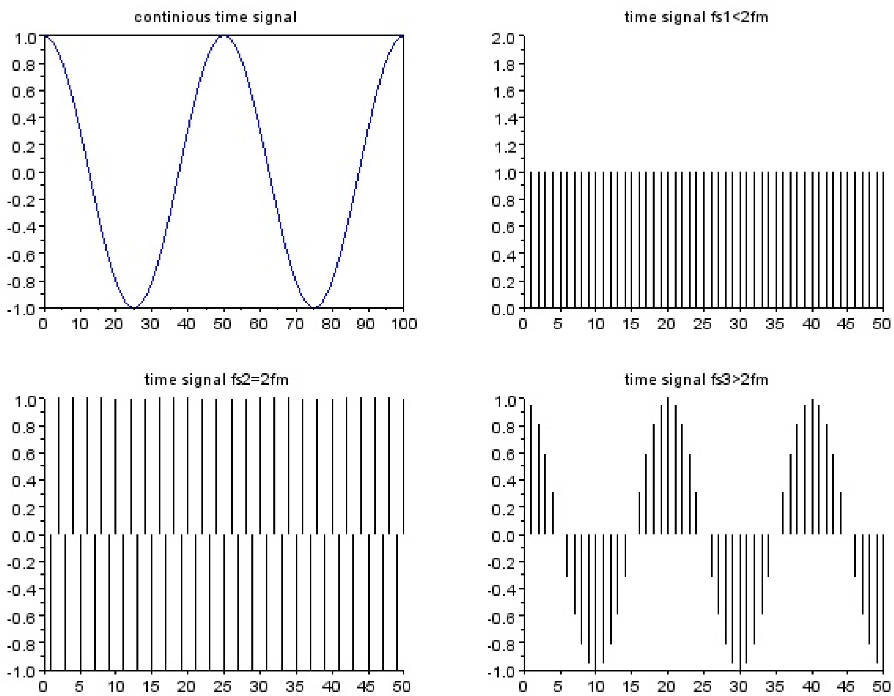

Figure 8.1: sampling

```
x2=cos(2*%pi*fm*n/fs2);
subplot(2,2,3);
plot2d3(n,x2);
title('time signal fs2=2fm');
fs3=0.4;
x3=cos(2*%pi*fm*n/fs3);
subplot(2,2,4);
plot2d3(n,x3);
title('time signal fs3>2fm');
```

Experiment: 9

PROGRAM FOR ALIASING PROCESS

Scilab code Solution 9.1 Aliasing process

```
//program for effect of aliasing
clc;
clf;
clear all;
fs=8000;
t=0:1/fs:0.01;
f=[1000,2000,3000,3500,4000,4500,8000,8500,9000];
i=1;
while i<7
    y=sin(2*%pi*f(i)*t);
    subplot(3,2,i);
    plot(t,y);
    xlabel('frequency');
    ylabel('amplitude');
    i=i+1;
end
```

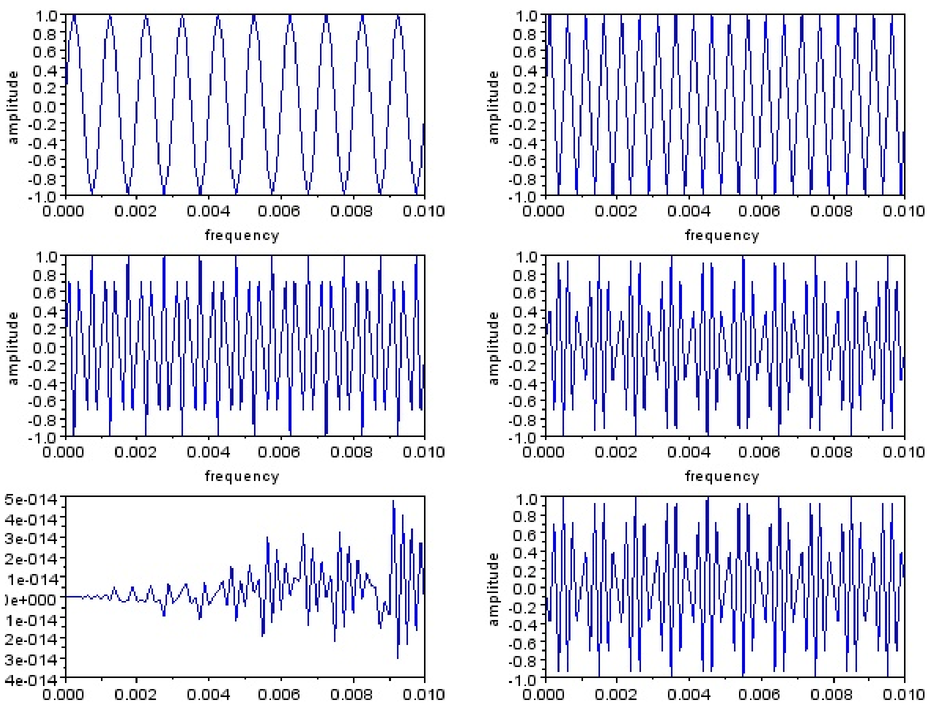

Figure 9.1: Aliasing process

Experiment: 10

LINEAR CONVOLUTION OF TWO SEQUENCES

Scilab code Solution 10.1 linear convolution of two sequences

```
//program for linear convolution
clc;
clf;
clear all;
x=input('Enter the first sequence:');
y=input('Enter the second Sequence:');
n=convol(x,y);
subplot(2,2,1);
plot2d3(x);
title('first sequence');
xlabel('n-------->');
ylabel('amp---->');
subplot(2,2,2);
plot2d3(y);
title('second sequence');
xlabel('n-------->');
ylabel('amp---->');
subplot(2,2,3);
plot2d3(n);
```

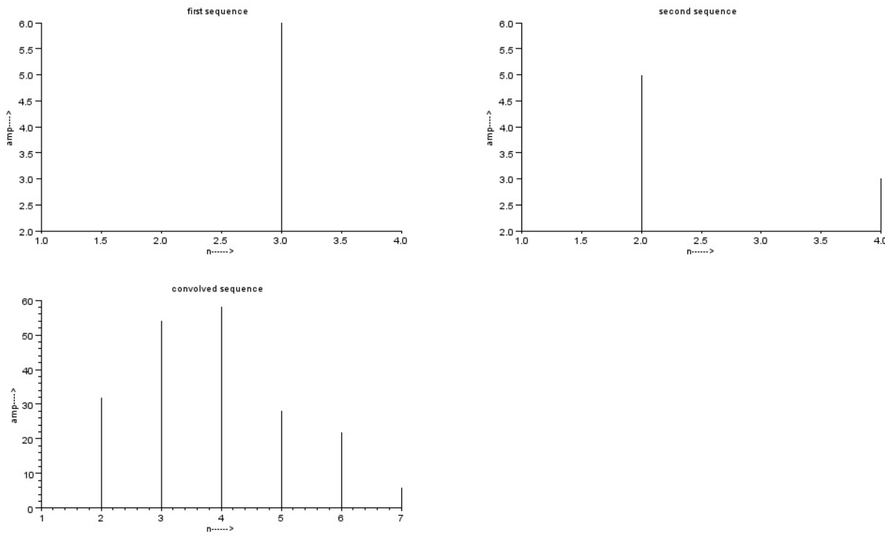

Figure 10.1: linear convolution of two sequences

```
title('convolved sequence');
xlabel('n------->');
ylabel('amp------>');
disp('The convolved sequence is');
disp(n);

//result:
//Enter the first sequence:[4 2 6 2]
//Enter the second Sequence:[6 5 2 3]
//
// The convolved sequence is
//
//     24.    32.    54.    58.    28.    22.    6.
```

Experiment: 11

CIRCULAR CONVOLUTION OF TWO SEQUENCES

Scilab code Solution 11.1 circular convolution

```
//program for circular convolution
clc;
clf;
clear all;
g=input('Enter the first sequence:');
h=input('Enter the second sequence:');
N1=length(g);
N2=length(h);
N=max(N1,N2);
N3=N1-N2;
if(N3>=0)  then
   h=[h,zeros(1,N3)];
else
    g=[g,zeros(1,-N3)];
end
for n=1:N
    y(n)=0;
```

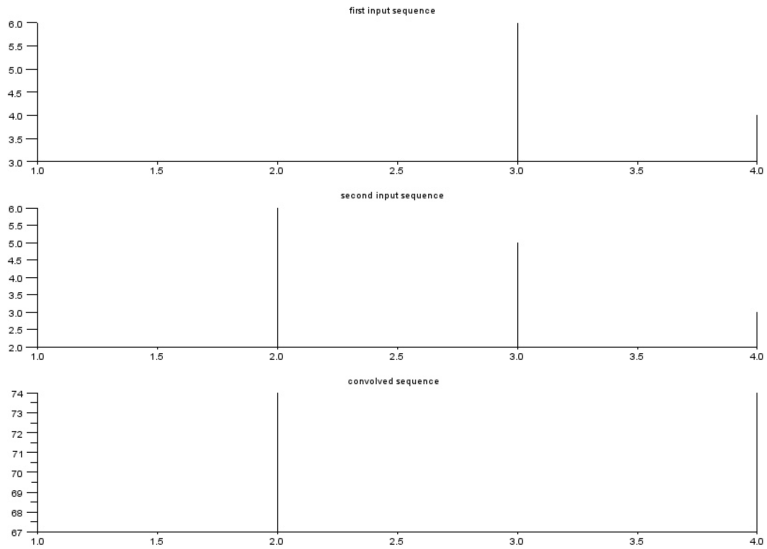

Figure 11.1: circular convolution

```
for i=1:N
        j=n-i+1;
        if(j<=0)
            j=N+j;
end
   y(n)=y(n)+g(i)*h(j);
end
end
disp('The resultant signal is:');
disp(y);
subplot(3,1,1);
plot2d3(g);
title('first input sequence');
subplot(3,1,2);
plot2d3(h);
title('second input sequence');
subplot(3,1,3);
plot2d3(y);
title('convolved sequence');

```

```
//result
//Enter the first sequence:[5 3 6 4]
//Enter the second sequence:[2 6 5 3]
//
// The resultant signal is:
//
//     73.     74.     67.     74.
```

Experiment: 12

DFT OF A SIGNAL

Scilab code Solution 12.1 DFT

```
//program for calculation of DFT of a signal
clc;
clf;
clear all;
N=input('Enter the value of N:');
x=input('enter input sequence');
for k=1:N
    y(k)=0;
    for n=1:N
        y(k)=y(k)+x(n).*exp(-%i*2*%pi*(k-1)*(n-1)/N)
           ;
        A=real(y);
        B=imag(y);
    end;
end;
mag=abs(y);
x1=atan(imag(y),real(y));
phase=x1*(180/%pi);
disp('the resultant DFT sequence is');
```

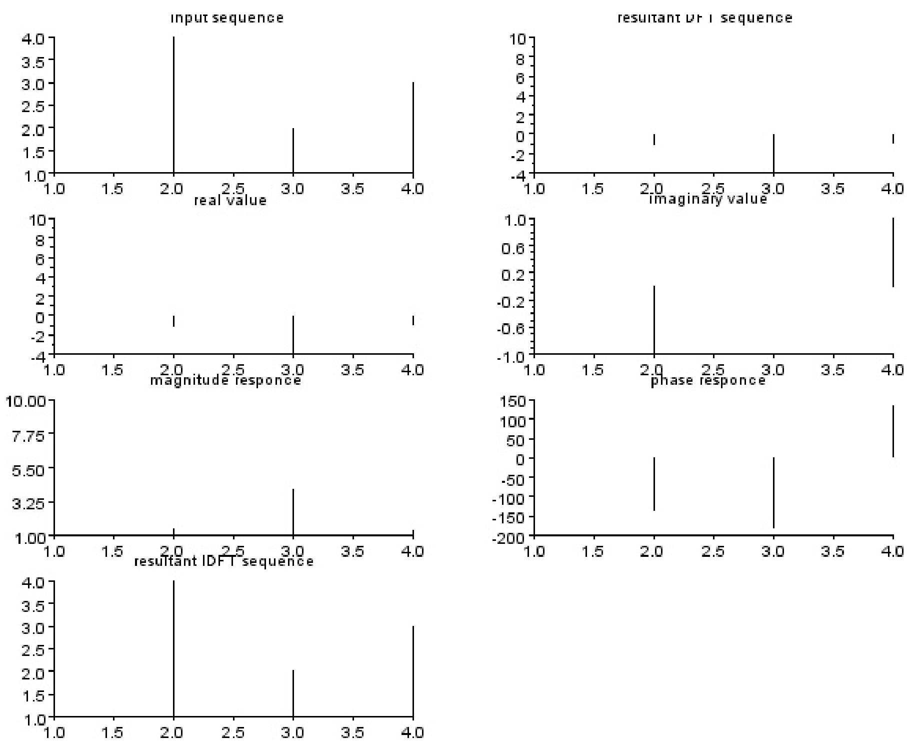

Figure 12.1: DFT

```
disp(y);
disp('the resultant real value is');
disp(A);
disp('the resultant imaginary value is');
disp(B);
disp('the magnitude response is');
disp(mag);
disp('the phase response is');
disp(phase);
for n=1:N
    y(n)=0;
    for k=1:N
        y(n)=y(n)+(1/N)*(y(k).*exp(%i*2*%pi*(k-1)*(n
           -1)/N));
        C=real(x);
    end;
end;
disp('the resultant IDFT sequence is');
disp(C);
subplot(4,2,1);
plot2d3(x);
title('input sequence');
subplot(4,2,2);
plot2d3(A);
title('resultant DFT sequence');
subplot(4,2,3);
plot2d3(A);
title('real value');
subplot(4,2,4);
plot2d3(B);
title('imaginary value');
subplot(4,2,5);
plot2d3(mag);
title('magnitude responce');
subplot(4,2,6);
plot2d3(phase);
title('phase responce');
subplot(4,2,7);
```

```
plot2d3(C);
title('resultant IDFT sequence');


//result
//Enter the value of N:4
//enter input sequence[1 4 2 3]

// the resultant DFT sequence is

//      10.   - 1. - i      - 4. - 1.102D-15i   - 1. + i

// the resultant real value is
//      10.   - 1.   - 4.   - 1.

// the resultant imaginary value is

//      0.   - 1.   - 1.102D-15     1.

// the magnitude response is

//      10.     1.4142136     4.     1.4142136

// the phase response is

//      0.   - 135.   - 180.     135.

// the resultant IDFT sequence is

//      1.     4.     2.     3.
```

Experiment: 13

FFT OF A SIGNAL

Scilab code Solution 13.1 fast fourier transform

```
//program for calculation of FFT of a signal
clc;
clf;
clear all;
N=input('Enter the value of N:');
x=input('enter input sequence');
y=fft(x);
A=real(y);
B=imag(y);
mag=abs(y);
x1=atan(imag(y),real(y));
phase=x1*(180/%pi);
disp('the resultant FFT sequence is');
disp(y);
disp('the magnitude response is');
disp(mag);
disp('the phase response is');
disp(phase);
z=ifft(y);
```

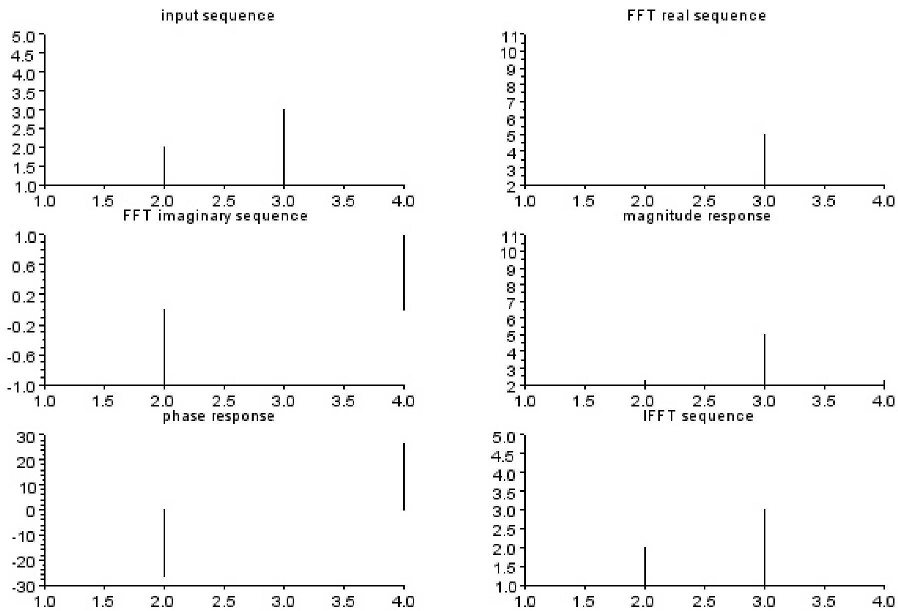

Figure 13.1: fast fourier transform

```
disp('the resultant IFFT sequence is');
disp(z);
subplot(3,2,1);
plot2d3(x);
title('input sequence');
subplot(3,2,2);
plot2d3(A);
title('FFT real sequence');
subplot(3,2,3);
plot2d3(B);
title('FFT imaginary sequence');
subplot(3,2,4);
plot2d3(mag);
title('magnitude response');
subplot(3,2,5);
plot2d3(phase);
title('phase response');
subplot(3,2,6);
plot2d3(x);
```

```
title('IFFT sequence');


//result
//Enter the value of N:4
//enter input sequence[5 2 3 1]

//the resultant FFT sequence is

//    11.    2. - i       5.    2. + i

//the magnitude response is

//    11.    2.236068    5.    2.236068

//the phase response is

//0.  - 26.565051    0.    26.565051

//the resultant IFFT sequence is

//    5.    2.    3.    1.
```