Scilab Textbook Companion for Signals And Systems by S. Sharma [1]

Created by
Shafquet Iftekhar
B.Tech (pursuing)
Electronics Engineering
Jamia Millia Islamia
College Teacher
Dr. Sajad A. Loan, JMI, New Delhi
Cross-Checked by
Giridharan, IITB

May 17, 2016

[1] Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Signals And Systems

Author: S. Sharma

Publisher: S. K. Kataria And Sons

Edition: 5

Year: 2006

ISBN: 81-85749-08-6

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

Chapter 1

Introduction to signals and systems

Scilab code Exa 1.1.a Check for periodicity

```
//Example 1.1a
//Determine whether the given signal is periodic or
   not
clc;
t=0:1/100:1
x=sin(15*%pi*t);
plot(x);
disp('ploting the signal and showing that it is
   periodic with period=2pi/15pi');
```

Scilab code Exa 1.1.b Check for periodicity

```
//Example 1.1b
//Determine whether the given signal is periodic or
   not
clc;
```

```
t=0:1/100:5
x=sin(sqrt(2)*%pi*t);
plot(x);
disp('ploting the signal and showing that itis
   periodic with period=2pi/sqrt(2)pi');
```

Scilab code Exa 1.4 Sketch and find power

```
//Example 1.4
//Sketch the signal x(t)=Asin(t)
clc;
A=0.5;
t=0:1/100:10
x=A*sin(t);
plot(x);
//since it is a periodic signal so it is power
   signal
P=(integrate('((0.5)^2)*(sin(t)^2)','t',0,2*%pi))
   /(2*%pi);
```

Scilab code Exa 1.5 Sketch and find energy

```
//Example 1.5
//Sketch the signal x(t)=A[u(t+a)-u(t-a)]
clc;
A=1;
a=2;
t=-a:a
x=1;
plot(t,x)
//this signal is a finite duration signal so it is
   energy signal
E=integrate('1','t',-a,a);
```

Scilab code Exa 1.6 Sketch and find energy

```
//Example 1.6
//Sketch the signal x(t)=exp(-a*t)
clc;
t=0:1/100:10;
x=exp(-0.5*t);
plot(x)
E=integrate('(exp(-0.5*t)^2)','t',0,10)
//Energy of the signal
```

Scilab code Exa 1.8 Find power of signal

```
//Example 1.8
//Find the power of the signal x(t)=Acos(Wot+theeta)
clc;
A=20;
Wo=(2*%pi)/4;
for i=1:50
    x(i)=A*cos(Wo*i);
end
p=0;
for i=1:4
    p=p+(abs(x(i)^2))/4;
end
disp(p,'The power of the given signal is =');
```

Scilab code Exa 1.14.a Check for causal system

```
//Example 1.14a
clc;
x=[1,2,3,4,0,4,3,2,1]
t=-length(x)/2:length(x)/2
count=0
mid=ceil(length(x)/2)
y=zeros(1,length(x))
y(mid+1:$)=x($:-1:mid+1)
for t=-1:-1:-mid
    y(t+1+mid)=x(-t)
end
for i=1:length(x)
    if(y(i)==x(i))
        count=count+1
     end
end
if(count==length(x))
    disp('THE GIVEN SYSTEM IS CAUSAL')
else
    disp('Since it depends on future values')
    disp('THE GIVEN SYSTEM IS NON CAUSAL')
end
```

Scilab code Exa 1.18.a Check for time invariant systems

```
//Example 1.18a
clc;
t0=1;
T=10;
for t=1:T
    x(t)=2*%pi*t/T;
    y(t)=sin(x(t));
end
inputshift=sin(x(T-t0));
outputshift=y(T-t0);
```

```
if(inputshift==outputshift)
    disp('THE GIVEN SYSTEM IS TIME INVARIANT')
else
    disp('THE GIVEN SYSTEM IS TIME VARIANT');
end
```

Scilab code Exa 1.18.b Check for time invariant systems

```
//Example 1.18b
clc;
t0=2;
T=10;
for t=1:T
    x(t)=t;
    y(t)=t*x(t);
end
inputshift=x(T-t0);
outputshift=y(T-t0);
if(inputshift==outputshift)
    disp('THE GIVEN SYSTEM IS TIME INVARIANT')
else
    disp('THE GIVEN SYSTEM IS TIME VARIANT');
end
```

Scilab code Exa 1.18.c Check for time invariant systems

```
//Example 1.18c
clc;
t0=2;
T=10;
for t=1:T
    x(t)=t;
    y(t)=x(t)*cos(200*%pi*t);
```

```
end
inputshift=x(T-t0);
outputshift=y(T-t0);
if(inputshift==outputshift)
    disp('THE GIVEN SYSTEM IS TIME INVARIANT')
else
    disp('THE GIVEN SYSTEM IS TIME VARIANT');
end
```

Scilab code Exa 1.19.a Check for linear systems

```
//Example 1.19a
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a=1
b=1
for t=1:length(x1)
    x3(t)=a*x1(t)+b*x2(t)
end
for t=1:length(x1)
    y1(t)=t*x1(t)
    y2(t)=t*x2(t)
    y3(t)=t*x3(t)
end
for t=1:length(y1)
    z(t)=a*y1(t)+b*y2(t)
end
count=0
for n=1:length(y1)
    if(y3(t)==z(t))
        count=count+1;
     end
end
if(count==length(y3))
```

```
    disp('It satisfy the superposition principle');
    disp('THE GIVEN SYSTEM IS LINEAR ');
else
    disp('It does not satisfy superposition
       principle ');
    disp('THE GIVEN SYSTEM IS NON LINEAR');
end
```

Scilab code Exa 1.19.b Check for linear systems

```
//Example 1.19b
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a=1
b=1
for t=1:length(x1)
    x3(t)=a*x1(t)+b*x2(t)
end
for t=1:length(x1)
    y1(t)=x1(t)^2
    y2(t)=x2(t)^2
    y3(t)=x3(t)^2
end
for t=1:length(y1)
    z(t)=a*y1(t)+b*y2(t)
end
count=0
for n=1:length(y1)
    if(y3(t)==z(t))
        count=count+1;
     end
end
if(count==length(y3))
disp('It satisfy the superposition principle');
```

```
disp('THE GIVEN SYSTEM IS LINEAR ');
else
    disp('It does not satisfy superposition
       principle ');
    disp('THE GIVEN SYSTEM IS NON LINEAR');
end
```

Scilab code Exa 1.20.b Check for linear systems

```
//Example 1.20b
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a=1
b=1
for t=1:length(x1)
    x3(t)=a*x1(t)+b*x2(t)
end
for t=1:length(x1)
    y1(t)=x1(t)^2
    y2(t)=x2(t)^2
    y3(t)=x3(t)^2
end
for t=1:length(y1)
    z(t)=a*y1(t)+b*y2(t)
end
count=0
for n=1:length(y1)
    if(y3(t)==z(t))
        count=count+1;
     end
end
if(count==length(y3))
disp('It satisfy the superposition principle');
disp('THE GIVEN SYSTEM IS LINEAR ');
```

```
else
    disp('It does not satisfy superposition
       principle ');
    disp('THE GIVEN SYSTEM IS NON LINEAR');
end
```

Scilab code Exa 1.21 Check for linear systems

```
//Example 1.21
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a1=1;
b1=1;
a=7;
b=5;
for t=1:length(x1)
    x3(t)=a1*x1(t)+b1*x2(t)
end
for t=1:length(x1)
    y1(t)=a*x1(t)+b
    y2(t)=a*x2(t)+b
    y3(t)=a*x3(t)+b
end
for t=1:length(y1)
    z(t)=a1*y1(t)+b1*y2(t)
end
count=0
for n=1:length(y1)
    if(y3(t)==z(t))
        count=count+1;
     end
end
if(count==length(y3))
disp('It satisfy the superposition principle');
```

```
disp('THE GIVEN SYSTEM IS LINEAR ');
else
    disp('It does not satisfy superposition
       principle ');
    disp('THE GIVEN SYSTEM IS NON LINEAR');
end
```

Scilab code Exa 1.22 Check for linear systems

```
//Example 1.22
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a1=1
b1=1
Wc=%pi
for t=1:length(x1)
    x3(t)=a1*x1(t)+b1*x2(t)
end
for t=1:length(x1)
    y1(t)=x1(t)*cos(Wc*t)
    y2(t)=x2(t)*cos(Wc*t)
    y3(t)=x3(t)*cos(Wc*t)
end
for t=1:length(y1)
    z(t)=a1*y1(t)+b1*y2(t)
end
count=0
for n=1:length(y1)
    if(y3(t)==z(t))
        count=count+1;
     end
end
if(count==length(y3))
disp('It satisfy the superposition principle');
```

```
disp('THE GIVEN SYSTEM IS LINEAR ');
else
    disp('It does not satisfy superposition
        principle ');
    disp('THE GIVEN SYSTEM IS NON LINEAR');
end
```

Scilab code Exa 1.25 Check for linear systems

```
//Example 1.25
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a1=1;
b1=1;
a=7;
b=3;
for t=1:length(x1)
    x3(t)=a1*x1(t)+b1*x2(t)
end
for t=1:length(x1)
    y1(t)=a*x1(t)+b
    y2(t)=a*x2(t)+b
    y3(t)=a*x3(t)+b
end
for t=1:length(y1)
    z(t)=a1*y1(t)+b1*y2(t)
end
count=0
for n=1:length(y1)
    if(y3(t)==z(t))
        count=count+1;
     end
end
if(count==length(y3))
```

```
disp('It satisfy the superposition principle');
disp('THE GIVEN SYSTEM IS LINEAR ');
else
    disp('It does not satisfy superposition
       principle ');
    disp('THE GIVEN SYSTEM IS NON LINEAR');
end
```

Scilab code Exa 1.27 Find energy of signal

```
//Example 1.27
//Energy of the signal x(t)=Aexp(-a*t).u(t)
clc;
A=2;
a=0.5;
E=integrate('(A*exp(-a*t))^2','t',0,100);//Energy of
    the given signal
```

Scilab code Exa 1.28 Find power of signal

```
//Example 1.28
//Power of the signal x(t)=A
clc;
A=2;
P=(integrate('A^2','t',0,100))/(2*100)
```

Scilab code Exa 1.30 Find energy of signal

```
//Example 1.30
```

```
//Detremine the energy of the signal x(n)=0.5^n for
   n >0
clc;
E=integrate('(0.5^n)','n',0,1000);
```

Scilab code Exa 1.31.a Check for periodicity

```
//Example 1.31a
//Determine whether the given signal is periodic or
   not
clc;
n=0:1/100:10
x=sin(6*%pi*n/7);
plot(x)//plotting the signal and showing it is
   periodic with period 2pi/(6pi/7);
```

Scilab code Exa 1.31.b Check for periodicity

```
//Example 1.31b
//Determine whether the given signal is periodic or
   not
clc;
n=0:1/1000:100
x=sin(n/8);
plot(x);//plotting the signal and showing that it is
    periodic with period 16pi
```

Scilab code Exa 1.33 Find power of signal

```
//Example 1.33
```

```
//Find the power of the signal x(t)=Acos(Wot+theeta)
clc;
A=10;
T=4;
Wo=(2*%pi)/T;
for i=1:T
    x(i)=A*cos(Wo*i);
end
p=0;
for i=1:T
    p=p+(abs(x(i)^2))/T;
end
disp(p,'The power of the given signal is =');
```

Scilab code Exa 1.34 Find energy of signal

```
//Example 1.34
//Find energy of x(t)=8exp(2+i4pi)t
clc;
E=0;
for t=1:100
    x(t)=8*exp((2+(%i*4*%pi))*t);
end
for t=1:100
    E=E+x(t)^2;
end
```

Scilab code Exa 1.39.a Sketch continous time signal

```
//Example 1.39a
//Sketch the signal x(t)=u(t)
clc;
t=0:1/100:10
```

```
x=1;
plot(t,x);
```

Scilab code Exa 1.39.b Sketch continous time signal

```
//Example 1.39b
//Sketch the signal x(t)=tu(t)
clc;
t=0:1/100:10
x=t
plot(t,x)
```

Scilab code Exa 1.43 Check for linear systems

```
//Example 1.43
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a=1
b=1
for t=1:length(x1)
    x3(t)=a*x1(t)+b*x2(t)
end
for t=1:length(x1)
    y1(t)=x1(t)^2
    y2(t)=x2(t)^2
    y3(t)=x3(t)^2
end
for t=1:length(y1)
    z(t)=a*y1(t)+b*y2(t)
end
count=0
for n=1:length(y1)
```

```
    if(y3(t)==z(t))
        count=count+1;
     end
end
if(count==length(y3))
disp('It satisfy the superposition principle');
disp('THE GIVEN SYSTEM IS LINEAR ');
else
    disp('It does not satisfy superposition
        principle ');
    disp('THE GIVEN SYSTEM IS NON LINEAR');
end
```

Scilab code Exa 1.47 Check for time invariant systems

```
//Example 1.47
clc;
k0=2;
n0=2;
N=10;
x=[1,2,3,4,5,6,7,8,9,10];
y=zeros(1,length(x));
for n=1:length(x)/k0
    y(n)=x(k0*n);
end
inputshift=x(N-n0);
outputshift=y(N-n0);
if(inputshift==outputshift)
    disp('THE GIVEN SYSTEM IS TIME INVARIANT')
else
    disp('THE GIVEN SYSTEM IS TIME VARIANT');
end
```

Scilab code Exa 1.49.a Check for periodicity

```
//Example 1.49a
//Determine whether the signal x(n)=sin(7/9*pi*(n^2)
   +1)
clc;
n=0:1/100:5
x=sin((7/9)*%pi*(n^2)+1)
plot(x);
disp('this shows that signal is NOT periodic');
```

Scilab code Exa 1.49.b Check for periodicity

```
//Example 1.49b
//Determine whether the signal x(n)=cos(pi*n/2)cos(
   pi*n/4)
clc;
n=0:1/100:100
x0=cos((%pi*n/2)+(%pi*n/4))
x1=cos((%pi*n/2)-(%pi*n/4))
x=(x0+x1)/2;
plot(x);
disp('plot shows that this is a periodic signal');
```

Chapter 2

Linear Time Invariant System

Scilab code Exa 2.1 Convolution of two continous time functions

```
//Example 2.1
clc;
t=-8:1/100:8;
for i=1:length(t)
   x(i)=exp(-t(i)^2);
   h(i)=3*t(i)^2;
end
y=convol(x,h);
figure
plot2d(t,h);
title('Impulse responce');
figure
plot2d(t,x);
title('Input signal');
figure
t2=-16:1/100:16
plot2d(t2,y);
title('Output signal');
```

Scilab code Exa 2.2 Find responce of system

```
//Example 2.2
clc;
t=-8:1/100:8;
for i=1:length(t)
    if t(i)<0 then
        x(i)=0;
        h(i)=0;
    else
         x(i)=exp(-3.*t(i));
         h(i)=1;
    end
end
t1=t+1;
y=convol(x,h);
figure
plot2d(t1,h);
title('Impulse responce');
figure
plot2d(t,x);
title('Input signal');
figure
t2=-16:1/100:16
plot2d(t2,y);
title('Output signal');
```

Scilab code Exa 2.3 Find unit step responce of system

```
//Example 2.3
clc;
R=100;
L=100;
t=-8:1/100:8;
for i=1:length(t)
```

```
    if t(i)<0 then
        x(i)=0;
        h(i)=0;
    else
         h(i)=(R/L)*exp(-(R/L).*t(i));
         x(i)=1;
    end
end
y=convol(x,h);
figure
plot2d(t,h);
title('Impulse responce');
figure
plot2d(t,x);
title('Input signal');
figure
t2=-16:1/100:16
plot2d(t2,y);
title('Output signal');
```

Scilab code Exa 2.4 Convolution of two continous time functions

```
//Example 2.4
clc;
t=-8:1/100:8;
for i=1:length(t)
         x(i)=3*cos(2.*t(i));
         h(i)=exp(-abs(t(i)));
end
y=convol(x,h);
figure
plot2d(t,h);
title('Impulse responce');
figure
plot2d(t,x);
```

```
title('Input signal');
figure
t2=-16:1/100:16
plot2d(t2,y);
title('Output signal');
```

Scilab code Exa 2.5 Evaluation of output of LTI system

```
//Example 2.5
clc;
Max_Limit=10;
h=ones(1,Max_Limit);
N2=0:length(h)-1;
a=0.5;//constant a>0
for t=1:Max_Limit
x(t)=exp(-a*(t-1));
end
N1=0:length(x)-1;
y=convol(x,h)-1;
N=0:length(x)+length(h)-2;
figure
a=gca();
plot2d(N2,h)
xtitle('Impulse Response','t','h(t)');
a.thickness=2;
figure
a=gca();
plot2d(N1,x)
xtitle('Input Response','t','x(t)');
a.thickness=2;
figure
a=gca ();
plot2d(N(1:Max_Limit),y(1:Max_Limit))
xtitle('Output Response','t','y(t)');
a.thickness=2;
```

Scilab code Exa 2.6 Find responce of system

```
//Example 2.6
clc;
t=-8:1/100:8;
for i=1:length(t)
    if t(i)<0 then
        x(i)=exp(2.*t(i));
        h(i)=0;
    else
         x(i)=0;
         h(i)=1;
    end
end
t1=t+3;
y=convol(x,h);
figure
plot2d(t1,h);
title('Impulse responce');
figure
plot2d(t,x);
title('Input signal');
figure
t2=-16:1/100:16
plot2d(t2,y);
title('Output signal');
```

Scilab code Exa 2.7 Convolution of two discrete time signals

```
//Example 2.7
clc;
```

```
n=-8:1:8;
for i=1:length(n)
         x(i)=exp(-n(i)^2);
         h(i)=3.*n(i)^2;
end
y=convol(x,h);
figure
plot2d3(n,h);
title('Impulse responce');
figure
plot2d3(n,x);
title('Input signal');
figure
n1=-16:1:16
plot2d3(n1,y);
title('Output signal');
```

Scilab code Exa 2.8 Find responce of system

```
//Example 2.8
clc;
n=-8:1:8;
for i=1:length(n)
    if n(i)<0 then
        x(i)=2^n(i);
        h(i)=0;
    else
         x(i)=0;
         h(i)=1;
    end
end
y=convol(x,h);
figure
plot2d3(n,h);
title('Impulse responce');
```

```
figure
plot2d3(n,x);
title('Input signal');
figure
n1=-16:1:16
plot2d3(n1,y);
title('Output signal');
```

Scilab code Exa 2.17.a Check for causal system

```
//Example 2.17a
clc;
disp('   y[n]=3x[n-2]+3x[n+2]   ');
disp('THE GIVEN SYSTEM IS NON-CAUSAL');
disp('Since the value of output depends on future
    input');
```

Scilab code Exa 2.17.b Check for causal system

```
//Example 2.17b
clc;
disp('   y[n]=x[n-1]+a*x[n-2]   ');
disp('THE GIVEN SYSTEM IS CAUSAL');
disp('Since the value of output doesnot depends on
    future input');
```

Scilab code Exa 2.17.c Check for causal system

```
//Example 2.17c
clc;
```

```
disp('    y[n]=x[-n]    ');
disp('THE GIVEN SYSTEM IS NON-CAUSAL');
disp('Since the value of output depends on future
    input');
```

Scilab code Exa 2.19.a Check for linear systems

```
//Example 2.19a
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a=1
b=1
a1=0.5
b1=0.5
for n=1:length(x1)
    x3(n)=a*x1(n)+b*x2(n)
end
for n=1:length(x1)
    y1(n)=a1*n*x1(n)+b1
    y2(n)=a1*n*x2(n)+b1
    y3(n)=a1*n*x3(n)+b1
end
for n=1:length(y1)
    z(n)=a*y1(n)+b*y2(n)
end
count=0
for n=1:length(y1)
    if(y3(n)==z(n))
        count=count+1;
     end
end
if(count==length(y3))
disp('It satisfy the superposition principle');
disp('THE GIVEN SYSTEM IS LINEAR ');
```

```
else
    disp('It does not satisfy superposition
       principle ');
    disp('THE GIVEN SYSTEM IS NON LINEAR');
end
```

Scilab code Exa 2.19.b Check for linear systems

```
//Example 2.19b
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a=1
b=1
for n=1:length(x1)
    x3(n)=a*x1(n)+b*x2(n)
end
for n=1:length(x1)
    y1(n)=exp(x1(n))
    y2(n)=exp(x2(n))
    y3(n)=exp(x3(n))
end
for n=1:length(y1)
    z(n)=a*y1(n)+b*y2(n)
end
count=0
for n=1:length(y1)
    if(y3(n)==z(n))
        count=count+1;
     end
end
if(count==length(y3))
disp('It satisfy the superposition principle');
disp('THE GIVEN SYSTEM IS LINEAR ');
else
```

```
    disp('It does not satisfy superposition
       principle ');
    disp('THE GIVEN SYSTEM IS NON LINEAR');
end
```

Scilab code Exa 2.21.a Check for linear systems

```
//Example 2.21
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a=1
b=1
for t=1:length(x1)
    x3(t)=a*x1(t)+b*x2(t)
end
for t=1:length(x1)
    y1(t)=5*sin(x1(t))
    y2(t)=5*sin(x2(t))
    y3(t)=5*sin(x3(t))
end
for t=1:length(y1)
    z(t)=a*y1(t)+b*y2(t)
end
count=0
for n=1:length(y1)
    if(y3(t)==z(t))
        count=count+1;
     end
end
if(count==length(y3))
disp('It satisfy the superposition principle');
disp('THE GIVEN SYSTEM IS LINEAR ');
else
    disp('It does not satisfy superposition
```

```
        principle ');
    disp('THE GIVEN SYSTEM IS NON LINEAR');
end
```

Scilab code Exa 2.21.b Check for linear systems

```
//Example 2.21b
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a=1
b=1
for t=1:length(x1)
    x3(t)=a*x1(t)+b*x2(t)
end
for t=1:length(x1)
    y1(t)=7*x1(t)+5
    y2(t)=7*x2(t)+5
    y3(t)=7*x3(t)+5
end
for t=1:length(y1)
    z(t)=a*y1(t)+b*y2(t)
end
count=0
for n=1:length(y1)
    if(y3(t)==z(t))
        count=count+1;
     end
end
if(count==length(y3))
disp('It satisfy the superposition principle');
disp('THE GIVEN SYSTEM IS LINEAR ');
else
    disp('It does not satisfy superposition
        principle ');
```

```
    disp('THE GIVEN SYSTEM IS NON LINEAR');
end
```

Scilab code Exa 2.25 Check for linear systems

```
//Example 2.25
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a=1
b=1
for n=1:length(x1)
    x3(n)=a*x1(n)+b*x2(n)
end
for n=1:length(x1)
        y1(n)=x1(n)^2
        y2(n)=x2(n)^2
        y3(n)=x3(n)^2
end
for n=1:length(y1)
    z(n)=a*y1(n)+b*y2(n)
end
count=0
for n=1:length(y1)
    if(y3(n)==z(n))
        count=count+1;
     end
end
if(count==length(y3))
disp('It satisfy the superposition principle');
disp('THE GIVEN SYSTEM IS LINEAR ');
else
    disp('It does not satisfy superposition
       principle ');
    disp('THE GIVEN SYSTEM IS NON LINEAR');
```

```
end
```

Scilab code Exa 2.59 Convolution of two discrete time signals

```
//Example 2.59
clc;
n=-8:1:8;
for i=1:length(n)
    if n(i)<0 then
        x(i)=0;
        h(i)=0;
    else
         x(i)=1;
         h(i)=2^n(i);
    end
end
y=convol(x,h);
figure
a=gca();
plot2d3(n,h);
a.x_location='origin';
a.y_location='origin';
title('Impulse responce');
figure
a=gca();
plot2d3(n,x);
a.x_location='origin';
a.y_location='origin';
title('Input signal');
figure
a=gca();
n1=-16:1:16
plot2d3(n1,y);
a.x_location='origin';
a.y_location='origin';
```

```
title('Output signal');
```

Chapter 3

Fourier Analysis of Periodic and APeriodic Continous Time Signal

Scilab code Exa 3.8 Fourier Transform

```
clc ;
close ;
// Analog S i g n a l
A =1; // Ampl i tude
Dt = 0.005;
t = 0: Dt :10;
xt = exp(-A*t);
Wmax = 2* %pi *1; // Analog Fr equency = 1Hz
K = 4;
k = 0:( K /1000) :K;
W = k* Wmax /K;
XW = xt* exp (- sqrt ( -1)*t'*W) * Dt;
XW_Mag = abs(XW);
W = [-mtlb_fliplr(W),W(2:1001)]; // Omega f rom
    Wmax to Wmax
XW_Mag = [mtlb_fliplr(XW_Mag),XW_Mag(2:1001)];
[ XW_Phase ,db] = phasemag (XW);
```

```
XW_Phase =[-mtlb_fliplr(XW_Phase),XW_Phase(2:1001)];
//Plotting Continuous Time Signal
figure
a = gca ();
a.y_location =" origin";
plot (t,xt);
xlabel ( ' t in sec. ' );
ylabel ( ' x ( t ) ' )
title ( ' Continuous Time Signal ' )
figure
// Pl o t t i n g Magni tude Re spons e o f CTS
subplot (2 ,1 ,1);
a = gca ();
a.y_location =" origin";
plot (W, XW_Mag );
xlabel ( ' Fr equency i n Radians /
    S e c o n d s         > W' );
ylabel ( ' abs (X(jW) ) ' )
title ( 'Magni tude Re spons e (CTFT) ' )
// Pl o t t i n g Phase Reponse o f CTS
subplot (2 ,1 ,2);
a = gca ();
a.y_location = " origin";
a.x_location = " origin";
plot (W, XW_Phase *%pi /180) ;
xlabel ( ' Fr equency in Radians /
    S e c o n d s         > W' );
ylabel ( '<X(jW) ' )
title ( ' Phase Re spons e (CTFT) i n Radians ' )
```

Scilab code Exa 3.9 Fourier Transform

```
//Example 3.9
clc;
clear;
```

```
A=1;
Dt=0.005;
T1=4;
t=-T1/2:Dt:T1/2;
for i=1:length(t)
xt(i)=A;
end
Wmax=2*%pi*1;
K=4;
k=0:(K/1000):K;
W=k*Wmax/K;
xt=xt';
XW=xt*exp(-sqrt(-1)*t'*W)*Dt;
XW_Mag=real(XW);
W=[-mtlb_fliplr(W),W(2:1001)];
XW_Mag=[mtlb_fliplr(XW_Mag),XW_Mag(2:1001)];
subplot(2,1,1);
a=gca();
a.data_bounds=[-4,0;4,2];
a.y_location="origin";
plot(t,xt);
xlabel('t in sec.');
title('Continous Time Signal x(t)');
subplot(2,1,2);
a=gca();
a.y_location="origin";
plot(W,XW_Mag);
xlabel('Frequency in Radians/Seconds');
title('Continuous time Fourier Transform X(jW)');
```

Scilab code Exa 3.15 Fourier Transform

```
//Example 3.15
clc;
clear;
```

```
T1=2;
T=4*T1;
Wo=2*%pi/T;
W=[-Wo,0,Wo];
ak=(2*%pi*Wo*T1/%pi)/sqrt(-1);
XW=[-ak,0,ak];
ak1=(2*%pi*Wo*T1/%pi);
XW1=[ak1,0,ak1];
figure
a=gca();
a.y_location="origin";
a.x_location="origin";
plot2d3('gnn',W,XW1,2);
poly1=a.children(1).children(1);
poly1.thickness=3;
xlabel('W');
title('CTFT of cos(Wot)');
```

Scilab code Exa 3.31 Fourier Transform

```
//Example 3.31
clc ;
clear;
R=10^3;
C=10^-3;
A=1/(R*C);
Dt=0.005;
t=0:Dt:10;
xt=A*exp(-A*t);
Wmax=2*%pi*1;
K=4;
k=0:(K/1000):K;
W=k*Wmax/K;
XW=xt*exp(-sqrt(-1)*t'*W)*Dt;
XW_Mag=abs(XW);
```

```
W=[-mtlb_fliplr(W),W(2:1001)];
XW_Mag=[mtlb_fliplr(XW_Mag),XW_Mag(2:1001)];
[XW_Phase,db]=phasemag(XW);
XW_Phase=[-mtlb_fliplr(XW_Phase),XW_Phase(2:1001)];
figure
a=gca();
a.y_location="origin";
plot(t,xt);
xlabel('t in sec.');
ylabel('x(t)');
title('Continuous Time Signal');
figure
subplot(2,1,1);
a=gca();
a.y_location="origin";
plot(W,XW_Mag);
xlabel('Frequency in Radians/Seconds>W');
ylabel('abs(X(jW))');
title('Magnitude Response (CTFT)');
subplot(2,1,2);
a=gca();
a.y_location="origin";
a.x_location="origin";
plot(W,XW_Phase*%pi/180) ;
xlabel(' Frequency in Radians/ Seconds        > W
    ' );
ylabel('<X(jW)')
title('Phase Response (CTFT) in Radians');
```

Chapter 4

The Discrete Time Fourier Transform

Scilab code Exa 4.1 DTFT computation

```
//Example 4.1
//Find the DTFT of (a^n)u[n],for |a|<1
clc;
syms w a n;
x=a^n;
X=symsum(x*exp(-%i*w*n),n,0,%inf);
```

Scilab code Exa 4.2 DTFT computation

```
//Example 4.2
//Find DTFT of x[n]=(a^n)u[-(n+1)]
clc;
syms w a n;
x=a^n;
X=symsum(x*exp(-%i*w*n),n,-%inf,-1);
```

Scilab code Exa 4.3 DTFT computation of unit impluse

```
//Example 4.3
//Find DTFT of unit impluse
clc;
syms w n;
x=1;
X=symsum(x*exp(-%i*w*n),n,0,0);
```

Scilab code Exa 4.5 DTFT computation

```
//Example 4.5
//Find DTFT of x[n]=a^|n| for -1<a<1
clc;
syms w a n;
x1=a^n;
x2=a^(-n);
X1=symsum(x1*exp(-%i*w*n),n,0,%inf);
X2=symsum(x2*exp(-%i*w*n),n,-%inf,-1);
X=X1+X2;
disp(X,'X(e^jw)=');
```

Scilab code Exa 4.9 Sketch discrete time signal

```
//Example 4.9
clc;
syms w a n;
x=a^n;
pi=22/7;
```

```
X=symsum(x*exp(-%i*w*n),n,0,%inf);
n1=0:10;
a=0.5;
x1=a^n1;
plot2d3(n1,x1);
xtitle('Discrete Time Signal','n','x[n]');
a.thickness=2;
```

Scilab code Exa 4.10 DTFT of cosine

```
//Example 4.10
//Find DTFT of x[n]=cos(Won) with Wo=(2*pi/5)
clc;
syms w n;
x1=exp(%i*(2*%pi*n/5));
x2=exp(-%i*(2*%pi*n/5));
X1=symsum(x1*exp(-%i*w*n),n,0,%inf);
X2=symsum(x2*exp(-%i*w*n),n,0,%inf);
X=(X1+X2)/2;
```

Scilab code Exa 4.12 DTFT of unit step

```
//Example 4.12
//Find DTFT of x[n]=u[n]
clc;
syms w n;
x=1;
X=symsum(x*exp(-%i*w*n),n,0,%inf);
```

Scilab code Exa 4.16 DTFT computation

```
//Example 4.16
//Find DTFT of x[n]=(a^n)u[n],for 0<a<1
clc;
syms w a n;
x=a^n;
X=symsum(x*exp(-%i*w*n),n,0,%inf);
```

Scilab code Exa 4.22 DTFT computation

```
//Example 4.22
//Find DTFT of x[n]=((1/2)^(n-1))u[n-1]
clc;
syms w n;
x=(1/2)^(n-1);
X=symsum(x*exp(-%i*w*n),n,1,%inf);
```

Chapter 5

Time and Frequency characterisation of signals and systems

Scilab code Exa 5.1 Bode Plot

```
//Example 5.1
//Obtain the Bode plot
clc;
s=%s;
H=syslin('c',2*10^4/(s^2+100*s+10^4));
bode(H,0.1,10000);
funcprot(0);
```

Scilab code Exa 5.2 Bode Plot

```
//Example 5.2
//Obtain the Bode plot
clc;
s=%s;
```

```
H=syslin('c',100*(1+s)/((10+s)*(100+s)));
bode(H,0.01,2000);
```

Chapter 6

Sampling And Laplace Transform

Scilab code Exa 6.1 Find nyquist rate

```
//Example 6.1
clc;
disp('x(t)=3cos(50%pi*t)+10sin(300%pi*t)-cos(100%pi*
   t)');
w1=50*%pi;
w2=300*%pi;
w3=100*%pi;
f1=w1/(2*%pi);
f2=w2/(2*%pi);
f3=w3/(2*%pi);
if f1>f2 then
    if f1>f3 then
        disp(2*f1,'Nyquist rate=');
    else
        disp(2*f3,'Nyquist rate=');
    end
else
    if f2>f3 then
        disp(2*f2,'Nyquist rate=');
```

```
    else
        disp(2*f3,'Nyquist rate=');
    end
end
```

Scilab code Exa 6.2 Find nyquist rate

```
//Example 6.2
clc;
disp('x(t)=(1/2%pi)cos(4000%pi*t)cos(1000%pi*t)');
w1=5000*%pi;
w2=3000*%pi;
f1=w1/(2*%pi);
f2=w2/(2*%pi);
if f1>f2 then
    nyquist_rate=2*f1;
else
    nyquist_rate=2*f2;
end
nyquist_interval=1/nyquist_rate;
disp(nyquist_rate,'Nyquist rate=');
disp(nyquist_interval,'Nyquist interval in seconds')
    ;
```

Scilab code Exa 6.4 Find nyquist rate

```
//Example 6.4
clc;
disp('x(t)=6cos(50%pi*t)+20sin(300%pi*t)-10cos(100
    %pi*t)');
w1=50*%pi;
w2=300*%pi;
w3=100*%pi;
```

```
f1=w1/(2*%pi);
f2=w2/(2*%pi);
f3=w3/(2*%pi);
if f1>f2 then
    if f1>f3 then
        disp(2*f1,'Nyquist rate=');
    else
        disp(2*f3,'Nyquist rate=');
    end
else
    if f2>f3 then
        disp(2*f2,'Nyquist rate=');
    else
        disp(2*f3,'Nyquist rate=');
    end
end
```

Scilab code Exa 6.26 Laplace transform of signal

```
//Example 6.26
//Find laplace transform x(t)=2e^(-3t)u(t)-e^(-2t)u(
   t)
clc;
syms t;
x=2*%e^(-3*t)-%e^(-2*t);
X=laplace(x);
```

Scilab code Exa 6.27.a Laplace transform of function

```
//Example 6.27a
//Laplace transform of x(t)=t^3+3*t^2-6*t+4
clc;
syms t;
```

```
x=t^3+3*t^2-6*t+4;
X=laplace(x);
```

Scilab code Exa 6.27.b Laplace transform of function

```
//Example 6.27b
//x(t)=(cos(3t))^3
clc;
syms t;
x=(cos(3*t))^3;
X=laplace(x);
```

Scilab code Exa 6.27.c Laplace transform of function

```
//Example 6.27c
clc;
syms a b t;
x=sin(a*t)*cos(b*t);
X=laplace(x);
```

Scilab code Exa 6.27.d Laplace transform of function

```
//Example 6.27d
clc;
syms t a;
x=t*sin(a*t);
X=laplace(x);
```

Scilab code Exa 6.27.e Laplace transform of function

```
//Example 6.27e
clc;
syms t s;
x1=1-%e^t;
X1=laplace(x1);
X=integ(X1,s,s,%inf);
```

Scilab code Exa 6.48 Find responce of system

```
//Example 6.48
clc;
syms t s;
x=1+%e^(-3*t)-%e^(-t);
X=laplace(x);
H=1/((s+1)*(s^2+s+1));
Y=X*H;
y=ilaplace(Y);
```

Chapter 7

The Z Transform

Scilab code Exa 7.1 z transform

```
//Example 7.1
clc;
syms a z n;
x=a^n;
X=symsum(x*(z^-n),n,0,%inf);
disp(X,'X(z)=');
```

Scilab code Exa 7.2 z transform of unit impulse

```
//Example 7.2
clc;
syms n z;
x=1;
X=symsum(x*(z^-n),n,0,0);
disp(X,'X(z)=');
```

Scilab code Exa 7.3 z transform of unit step

```
clc;
syms n;
x=ones(1);
X=symsum(x*(z^-n),n,0,%inf);
disp(X,'X(z)=');
```

Scilab code Exa 7.5 z transform of cosine

```
//Example 7.5
clc;
syms Wo n z;
x1=exp(sqrt(-1)*Wo*n);
X1=symsum(x1*(z^-n),n,0,%inf);
x2=exp(-sqrt(-1)*Wo*n);
X2=symsum(x2*(z^-n),n,0,%inf);
X=(X1+X2)/2;
disp(X,'X(z)=');
```

Scilab code Exa 7.6 z transform

```
//Example 7.6
clc;
syms n z;
x=1;
X=symsum(x*(z^-n),n,-%inf,0);
disp(X,'X(z)=');
```

Scilab code Exa 7.7 z transform of sequence

```
//Example 7.7
clc;
syms n z;
X1=0;
X2=0;
for i=0:2:4
    x1=(1/2)^i;
    X1=X1+x1*z^-i;
end
for i=1:2:5
    x2=(1/3)^i;
    X2=X2+x2*z^-i;
end
x3=2^n;
X3=symsum(x3*(z^-n),n,-%inf,1);
X=X1+X2+X3;
disp(X,'X(z)=');
```

Scilab code Exa 7.10 z transform

```
//Example 7.10
//Z-transform of (2^n)u[n-2]
clc;
syms n z;
x=2^n;
X=symsum(x*(z^-n),n,2,%inf);
disp(X,'X(z)=');
```

Scilab code Exa 7.11.a z transform

```
//Example 7.11a
//Z transform of (a^n)cos(Wo*n)
clc;
```

```
syms Wo n z a;
x1=(a^n)*exp(sqrt(-1)*Wo*n);
X1=symsum(x1*(z^-n),n,0,%inf);
x2=(a^n)*exp(-sqrt(-1)*Wo*n);
X2=symsum(x2*(z^-n),n,0,%inf);
X=(X1+X2)/2;
disp(X,'X(z)=');
```

Scilab code Exa 7.11.b z transform

```
//Example 7.11b
//Z transform of (a^n)sin(Wo*n)
clc;
syms Wo n z a;
x1=(a^n)*exp(sqrt(-1)*Wo*n);
X1=symsum(x1*(z^-n),n,0,%inf);
x2=(a^n)*exp(-sqrt(-1)*Wo*n);
X2=symsum(x2*(z^-n),n,0,%inf);
X=(X1-X2)/(2*%i);
disp(X,'X(z)=');
```

Scilab code Exa 7.12 z transform using differentiation property

```
//Example 7.12
//Ztransform of x[n]=(n^2)*u[n] done by
   Diffrentiation property
clc;
syms z n;
x=1;
X1=symsum(x*(z^-n),n,0,%inf);
X2=(-z)*(diff(X1,z));
X=(-z)*(diff(X2,z));
disp(X,'X(z)=');
```

Scilab code Exa 7.13 z transform

```
//Example 7.13
//Diffrentiation property is used here
clc;
syms n z;
x1=((-1/2)^n);
x2=(1/4)^-n;
X1=symsum(x1*(z^-n),n,0,%inf);
X3=(-z)*diff(X1,z);
X2=symsum(x2*(z^-n),n,-%inf,0);
X=X3*X2;
disp(X,'X(z)=');
```

Scilab code Exa 7.18 Find Discrete time input signal

```
//Example 7.18
//Determine the input x[n] if h[n]=[1,2,3]and y[n
    ]=[1,1,2,-1,3]
clc;
clear;
function[za]=ztransfer(sequence,n)
    z=poly(0,'z','r')
    za=sequence*(1/z)^n'
endfunction
z=poly(0,'z');
h=[1,2,3];
n1=0:length(h)-1;
H=ztransfer(h,n1);
y=[1,1,2,-1,3];
n2=0:length(y)-1;
```

```
Y=ztransfer(y,n2);
X=Y/H;
funcprot(0);
funcprot(0);
x=ldiv(z^2-z+1,z^2,3);
disp(x,'x[n]=');
```

Scilab code Exa 7.19 Inverse Z transform

```
//Example 7.19
//Find the inverse Z-transform using long division
   method
clc;
clear;
z=poly(0,'z');
x=ldiv(z^2,z^2-(3/2)*z+(1/2),4);
disp(x,'x[n]=');
```

Scilab code Exa 7.24.a Inverse Z transform using long division method

```
//Example 7.24a
//Inverse Z-transform using long division method
clc;
clear;
z=poly(0,'z');
x=ldiv(2*z^3+3*z^2,(z+1)*(z+0.5)*(z-0.25),4);
disp(x,'x[n]=');
```

Scilab code Exa 7.36 z transform

```
//Example 7.36
clc;
syms z n;
x1=2^n;
X1=symsum(x1*(z^-n),n,0,%inf);
x2=3^n;
X2=symsum(x2*(z^-n),n,0,%inf);
X=3*X1-4*X2;
```

Scilab code Exa 7.37 z transform

```
//Example 7.37
clc;
syms z n;
x=(1/2)^n;
X=symsum(x*(z^-n),n,0,%inf);
```

Scilab code Exa 7.38 z transform

```
//Example 7.38
clc;
syms a z n;
x=-(a^n);
X=symsum(x*(z^-n),n,-%inf,-1);
```

Scilab code Exa 7.39 z transform using differentiation property

```
//Example 7.39
clc;
syms z n a;
```

```
x1=(a^n);
X1=symsum(x1*(z^-n),n,0,%inf);
X=(-z)*(diff(X1,z));
```

Scilab code Exa 7.42.a z transform of sequence

```
//Example 7.42a
clc;
function[za]=ztransfer(sequence,n)
    z=poly(0,'z','r')
    za=sequence*(1/z)^n'
endfunction
x=[1,2,3,4,5,0,7];
n1=0:length(x)-1;
X=ztransfer(x,n1);
funcprot(0);
```

Scilab code Exa 7.42.b z transform of sequence

```
//Example 7.42b
clc;
function[za]=ztransfer(sequence,n)
    z=poly(0,'z','r')
    za=sequence*(1/z)^n'
endfunction
x=[1,2,3,4,5,0,7];
n1=-3:length(x)-4;
X=ztransfer(x,n1);
funcprot(0);
```

Scilab code Exa 7.43 z transform

```
//Example 7.43
clc;
syms z n;
x1=(-1/3)^n;
x2=(1/2)^n;
X1=symsum(x1*(z^-n),n,0,%inf);
X2=symsum(x2*(z^-n),n,-%inf,-1);
X=X1-X2;
```

Scilab code Exa 7.48.a z transform

```
//Example 7.48a
clc;
syms z n;
x1=2^n;
X1=symsum(x1*(z^-n),n,0,%inf);
x2=(1/2)^n;
X2=symsum(x2*(z^-n),n,0,%inf);
X=X1+(3*X2);
```

Scilab code Exa 7.48.b z transform

```
//Example 7.48b
clc;
syms z n;
x1=(-1/2)^n;
X1=symsum(x1*(z^-n),n,0,%inf);
x2=(3)^n;
X2=symsum(x2*(z^-n),n,-%inf,-1);
X=(3*X1)-(2*X2);
```

Scilab code Exa 7.50 z transform of sequence

```
//Example 7.50
clc;
syms n z;
X1=0;
X2=0;
for i=0:2:4
    x1=(1/2)^i;
    X1=X1+x1*z^-i;
end
for i=1:2:5
    x2=(1/3)^i;
    X2=X2+x2*z^-i;
end
x3=(2)^n;
X3=symsum(x3*(z^-n),n,-%inf,1);
X=X1+X2+X3;
```

Scilab code Exa 7.52 z transform of discrete signal

```
//Example 7.52
//Z transform of x[n]=(2^n)u[n-2]
clc;
syms z n;
x=2^n;
X=symsum(x*(z^-n),n,2,%inf);
```

Scilab code Exa 7.54 Inverse Z transform

```
//Example 7.54
clc;
clear;
z=poly(0,'z');
X=[2;3*z^-1;4*z^-2];
n=0:2;
ZI=z^n';
x=numer(X.*ZI);
disp(x,'x[n]=');
```

Scilab code Exa 7.56 Find Discrete time input signal

```
//Example 7.56
//Determine the input x[n] if h[n]=[1,2,3]and y[n
    ]=[1,1,2,-1,3]
clc;
clear;
function[za]=ztransfer(sequence,n)
    z=poly(0,'z','r')
    za=sequence*(1/z)^n'
endfunction
z=poly(0,'z');
h=[1,2,3];
n1=0:length(h)-1;
H=ztransfer(h,n1);
y=[1,1,2,-1,3];
n2=0:length(y)-1;
Y=ztransfer(y,n2);
X=Y/H;
funcprot(0);
funcprot(0);
x=ldiv(1-z+z^2,z^2,3);
disp(x,'x[n]=');
```

Scilab code Exa 7.59.a z transform

```
//Example 7.59a
//Z transform of x[n]=-(a^n)u[-n-1]
clc;
syms a n z;
x=-(a^n);
X=symsum(x*(z^-n),n,-%inf,-1);
```

Scilab code Exa 7.59.b z transform

```
//Example 7.59b
//Z transform of x[n]=(a^-n)u[-n-1]
clc;
syms a n z;
x=(a^-n);
X=symsum(x*(z^-n),n,-%inf,-1);
```

Scilab code Exa 7.61.a z transform of discrete signal

```
//Example 7.61a
clc;
syms z n;
x1=(1/2)^n;
X1=symsum(x1*(z^-n),n,0,%inf);
x2=(1/3)^n;
X2=symsum(x2*(z^-n),n,0,%inf);
X=X1+X2;
```

Scilab code Exa 7.61.b z transform of discrete signal

```
//Example 7.61b
clc;
syms z n;
x1=(1/3)^n;
X1=symsum(x1*(z^-n),n,0,%inf);
x2=(1/2)^n;
X2=symsum(x2*(z^-n),n,-%inf,-1);
X=X1+X2;
```

Scilab code Exa 7.61.c z transform of discrete signal

```
//Example 7.61c
clc;
syms z n;
x1=(1/2)^n;
X1=symsum(x1*(z^-n),n,0,%inf);
x2=(1/3)^n;
X2=symsum(x2*(z^-n),n,-%inf,-1);
X=X1+X2;
```

Scilab code Exa 7.65 z transform

```
//Example 7.65
clc;
syms z n;
h1=(1/2)^n;
H1=symsum(h1*(z^-n),n,0,%inf);
```

```
h2=(-1/4)^n;
H2=symsum(h2*(z^-n),n,0,%inf);
H=(H1+H2)/2;
```

Scilab code Exa 7.68.a z transform of discrete signal

```
//Example 7.68a
//Z transform of x[n]=u[n]
clc;
syms n z;
x=1;
X=symsum(x*(z^-n),n,0,%inf);
```

Scilab code Exa 7.68.b z transform of discrete signal

```
//Example 7.68b
//Z transform of x[n]=-u[-n-1]
clc;
syms n z;
x=-1;
X=symsum(x*(z^-n),n,-%inf,-1);
```

Chapter 8

Discrete Fourier Transform and Fast Fourier Transform

Scilab code Exa 8.1 Convolution of two finite duration sequences

```
//Example 8.1
//Determine the convolution of the two finite
    duration sequence
clc;
x=[1,1,1];
n1=-1:1;
h=[1,1,1];
n2=-1:1;
y=convol(x,h);
n=-2:1:2;
disp(y,'y[n]=');
a = gca ();
a.y_location ="origin";
a.x_location ="origin";
plot2d3(n,round(y),5);
poly1=a.children(1).children (1);
poly1.thickness=2;
xtitle('Plot of sequence y[n]','n','y[n]');
funcprot(0);
```

Scilab code Exa 8.2 Responce of an FIR filter

```
//Example 8.2
//Find the response of an FIR filter with impulse
   response h[n]=[1,2,4] //to the input sequence x[n
   ]=[1,2]
clc;
x=[1,2];
h=[1,2,4];
Y=convol(x,h);
disp(Y,'y[n]=');
```

Scilab code Exa 8.3 DFT and IDFT

```
//Example 8.3
//Compute DFT of x(n)={1,1,0,0} and IDFT of y(n)
   ={1,0,1,0}
clc;
x=[1,1,0,0];
Y=[1,0,1,0];
X=fft(x,-1);
y=fft(Y,1);
```

Scilab code Exa 8.4 DFT computation

```
//Example 8.4
//Compute DFT of the following sequence
clc;
x=[0.25,0.25,0.25];
```

```
X=fft(x,-1);
disp(X,'X[k]=');
```

Scilab code Exa 8.5 DFT computation

```
//Example 8.5
//Find the DFT of the following sequence
clc;
x=[0.2,0.2,0.2];
n=-1:1;
X=fft(x,-1);
disp(X,'X[k]=');
```

Scilab code Exa 8.6 DFT of sequence

```
//Example 8.6
//Determine the DFT of the following sequence
clc;
x=[1,1,2,2,3,3];
X=fft(x,-1);
disp(X,'X[k]=');
```

Scilab code Exa 8.7 DFT computation

```
//Example 8.7
//DFT of x[n]=a.^n
clc;
a=0.5;//Say for a=0.5
n=0:4;
x=a.^n;
```

```
X=fft(x,-1);
disp(X,'X[k]=');
```

Scilab code Exa 8.8 DFT computation

```
//Example 8.8
//Compute 4-point DFT of the sequence x[n]=cos(n*pi
    /4)
clc;
n=0:3;
pi=22/7;
x=cos(n*pi/4);
X=fft(x,-1);
disp(X,'X[k]=');
```

Scilab code Exa 8.9 IDFT computation

```
//Example 8.9
//Computing IDFT of the following sequence
clc;
X=[1,2,3,4];
x=fft(X,1);
disp(x,'x[n]=');
```

Scilab code Exa 8.10 IDFT computation

```
//Example 8.10
//Find the IDFT of the following sequence
clc;
i=sqrt(-1);
```

```
X=[3,2+i,1,2-i];
x=fft(X,1);
disp(x,'x[n]=');
```

Scilab code Exa 8.11 DFT computation using FFT algorithm

```
//Example 8.11
//For the given x[n] determine X[k] using FFT
    algorithm
clc;
x=[1,2,3,4,4,3,2,1];
X=fft(x,-1);
disp(X,'X[k]=');
```

Scilab code Exa 8.12 DFT computation using FFT algorithm

```
//Example 8.12
//For the given x[n] determine X[k] using FFT
    algorithm
clc;
x=[0,1,2,3,4,5,6,7];
X=fft(x,-1);
disp(X,'X[k]=');
```

Scilab code Exa 8.13 DFT computation

```
//Example 8.13
//Find the DFT of the following sequence
clc;
h=[1/3,1/3,1/3];
```

```
H=fft(h,-1);
disp(H,'H[k]=');
```

Scilab code Exa 8.14 DFT computation

```
//Example 8.14
//Find the DFT of the following sequence
clc;
h=[1/3,1/3,1/3];
H=fft(h,-1);
disp(H,'H[k]=');
```

Scilab code Exa 8.15 DFT computation

```
//Example 8.15
//Obtain the DFT of x[n]=(a^n).u[n]
clc;
a=0.5;
for i=0:1:7
    x(i+1)=a.^i;
end
X=fft(x,-1);
disp(X,'X[k]=');
```

Scilab code Exa 8.16 DFT computation

```
//Example 8.16
//Program to find DFT of sequence x
   =[1,1,1,1,1,1,1,0]
clc ;
```

```
x=[1,1,1,1,1,1,1,0];
X=fft(x,-1);
disp(X,'X[k]=');
```

Scilab code Exa 8.17 DFT computation

```
//Example 8.17
//Determine the DFT of the following sequence
clc;
x=[0,0,1,1,1,1,1,0,0,0];
X=fft(x,-1);
disp(X,'X[k]=');
```