Scilab Textbook Companion for Signals And Systems by T. Rawat[1]

Created by
Payal Pawar
BE
Electrical Engineering
Government COllege Of Engineering,Karad
College Teacher
None
Cross-Checked by
None

August 23, 2018

[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: T. Rawat

Publisher: Oxford University Press, New Delhi

Edition: 1

Year: 2010

ISBN: 0198066791

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 system

Scilab code Exa 1.2.a sketch the signal

```
//example 1_2<a>
//sketch the following signal x(3t)
clc;
clear all;
t=-1/3:0.0001:1/3;
for i=1:length(t)
if t(i)<0 then
x(i)=1+3*t(i);
else
x(i)=1-3*t(i);
end
end
plot2d(t,x)
plot (t,x, 'red' );
xtitle('required figure','t','x(3*t)');
xgrid();
```

Scilab code Exa 1.2.b sketch the signal

```
//example 1_2<b>
//sketch the following signal x(3t+2)
clc;
clear all;
t=-1:0.0001:-1/3;
for i=1:length(t)
if t(i)<-2/3 then
x(i)=3+3*t(i);
else
x(i)=-1-3*t(i);
end
end
plot(t,x)
plot(t,x, 'red' );
xtitle('required figure','t','x(3*t+2)');
xgrid();
```

Scilab code Exa 1.2.c sketch the signal

```
//example 1_2<c>
//sketch the following signal x(-2t-1)
clc;
clear all;
t=-1:0.0001:0;
for i=1:length(t)
if t(i)>=-1/2 then
x(i)=-2*t(i);
else
x(i)=(2*t(i))+2;
end
end
plot(t,x)
plot(t,x, 'red' );
```

```
xtitle('required figure','t','x(-2*t-1)');
xgrid();
```

Scilab code Exa 1.6.a draw the waveforms of the siganals

```
//example 1.6<a>
//draw the waveform of the signal x1(t)=u(t+2)
clc ;
clear all;
t=-10:.001:10;
for i=1:length(t)
if t(i)>=-2 then
x(i)=1;
else
x(i)=0;
end
end
plot2d(t,x);
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.6.b draw the waveforms of the siganals

```
//draw the waveform of the signal x2(t)=u(t-2)
clc;
clear all;
t=-10:.001:10;
for i=1:length(t)
if t(i)>=2 then
x(i)=1;
else
x(i)=0;
end
end
```

```
//figure
plot2d(t,x);
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.6.c draw the waveforms of the siganals

```
//draw the waveform of the signal x3(t)=u(-t)
clc ;
clear all;
t=-10:.001:10;
for i=1:length(t)
if t(i)<=0 then
x(i)=1;
else
x(i)=0;
end
end
plot2d(t,x);
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.6.d draw the waveforms of the siganals

```
//draw the waveform of the signal x4(t)=u(-2t+1)
clc ;
clear all;
t=-10:.001:10;
for i=1:length(t)
if t(i)<=1/2 then
x(i)=1;
else
x(i)=0;
end
end
```

```
plot2d(t,x);
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.6.e draw the waveforms of the siganals

```
clc ;
clear all;
t=-10:.001:10;
for i=1:length(t)
if t(i)<=-1/2 then
x(i)=1;
else
x(i)=0;
end
end
// f i g u r e
f=scf(0);
plot2d(t,x);
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.6.f draw the waveforms of the siganals

```
//Example 1.6<f>
//draw the waveform of the signal x6(t)=u(t+2)-u(t
   -2)
clc ;
clear all;
t=-10:.001:10;
for i=1:length(t)
if t(i)>=-2 & t(i)<=2 then
x(i)=1;
else
x(i)=0;
```

```
end
end
//figure
plot2d(t,x);
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.6.g draw the waveforms of the siganals

```
//Example 1.6<g>//
//draw the waveform of the signal x7(t)=u(t)-2*u(t
    -1)+u(t-2)//
clc ;
clear all;
t=-10:0.001:10
for i=1:length(t)
if t(i)>=0 & t(i)<=1 then
x(i)=1;
else if t(i)<1 & t(i)>2 then
x(i)=0
end
end
if t(i)>1 & t(i)<=2 then
x(i)=-1
end
end
//figure
plot2d(t,x);
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.7.a draw the waveforms of the siganals

```
//Example 1.7<a>
//draw the waveform of the signal x1(t)=r(t-1)
```

```
clc;
clear all;
t=-10:.001:10;
for i=1:length(t)
if t(i)>=0 then
x(i)=t(i)+1;
else
x(i)=0;
end
end
plot2d(x,t);
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.7.b draw the waveforms of the siganals

```
//Example 1.7<b>
//draw the waveform of the signal x1(t)=r(t+1)
clc;
clear all;
t=-10:.001:10;
for i=1:length(t)
if t(i)>=-1 then
x(i)=t(i)+1;
else
x(i)=0;
end
end
plot2d(t,x);
xtitle('Required figure','x(t)','t')
```

Scilab code Exa 1.7.c draw the waveforms of the siganals

```
//Example 1.7<c>
```

```
//draw the waveform of the signal x1(t)=r(-t)
clc;
clear all;
t=-10:0.001:10;
for i=1:length(t)
if t(i)>=0  then
x(i)=-t(i);
else
x(i)=0;
end
end
//figure
plot2d(x,t);
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.7.d draw the waveforms of the siganals

```
//Example 1.7<d>
//draw the waveform of the signal x1(t)=r(3t)
clc;
clear all;
t=-10:.001:10;
for i=1:length(t)
if t(i)>=0 then
x(i)=3*t(i);
else
x(i)=0;
end
end
//figure
plot2d(x,t);
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.7.e draw the waveforms of the siganals

```
//Example 1.7<e>
//draw the waveform of the signal x1(t)=r(−3t)
clc;
clear all;
t=-10:.001:10;
for i=1:length(t)
if t(i)>=0 then
x(i)=-3*t(i);
else
x(i)=0;
end
end
//figure
plot2d(x,t);
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.7.f draw the waveforms of the siganals

```
//Example 1.7<f>
//draw the waveform of the signal x1(t)=r(3t+1)
clc;
clear all;
t=-10:.001:10;
for i=1:length(t)
if t(i)>=-1/3 then
x(i)=3*t(i)-1/3;
else
x(i)=0;
end
end
//figure
plot2d(x,t);
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.8.a draw the waveforms of the siganals

```
//Example 1.8<a>
//draw the waveform of the signal x1(t)=r(t)-r(t-1)-
   u(t-1)
clc;
clear all;
t=-10:0.001:10;
for i=1:length(t)
if t(i)>=0 & t(i)<1 then
x(i)=t(i);
else
x(i)=0;
end
end
plot2d(t,x)
xtitle('Required figure','t','x(t)')
```

Scilab code Exa 1.14.a draw the waveforms of the siganals

```
//Example 1.14<a>
//check the given signL is PERIODIC OR NOT//
clc ;
t=-10:.01:10;
x=%i*(exp(%i*10*t));
subplot (311)
plot (t,x);
disp ('(a) this shows that the given signal is
   periodic with period (.2*%pi)');
```

Scilab code Exa 1.14.b determine whether the signal is periodic or not

```
//Example 1.14<b>
//check the given signL is PERIODIC OR NOT e^(-1+j)*
    t
clc ;
t=-1:.01:1;
x=(exp(-1+%i)*t);
f=1/2*%pi;
N=1/f;
disp (N,'this shows that the given signal is not
    periodic');
```

Scilab code Exa 1.14.c determine whether the signal is periodic or not

```
//Example 1.14<c>
// Find wheather the following signal is periodic or
     not x3(n)=e^(i*7*pi*n)
clc;
n=-21:21;
x=exp(%i *7* %pi *n);
f=(7*%pi)/(2*%pi);
N=1/f;
disp(N,'the given signal is periodic');
```

Scilab code Exa 1.15.a determine whether the signal is periodic or not

```
//Example 1.15<a>
// Find wheather the following signal is periodic or
     not x3(n)=2*e^(%i*(t+%pi/4))
clc;
t=-21:21;
x=2*exp(%i*(t+%pi/4));
```

```
f=1/(2*%pi);
N=1/f;
disp('samples',N,'(b)the given signal is not
   periodic');
```

Scilab code Exa 1.18.c determine values of power and energy

```
// example 1.18<c>
//determine the values of power and energy
clc ;
t =0:0.01:100;
A=1;
x=A*cos (t);
P=(integrate('(A*cos(t))^2','t',0,2*%pi ))/(2*%pi);
disp(P,'The power of the signal is:');
E=(integrate('(A*cos(t))^2','t',0,2*%pi));
disp(E,'The energy is:');
disp('As t tends to infinity energy also tends to
   infinity but power remains finite hence it is
   power signal');
```

Scilab code Exa 1.18.d determine values of power and energy

```
//Example 1.18<d>
//determine the values of power and energy
clc ;
E=0;
for n=0:200
x(n+1)=(1/2)^n;
end
for n=0:200
E=E+x(n +1)^2;
end
```

```
disp(E,'The energy of the signal is; ' );
disp ('since the energy is finite , hence it is
   energy signal');
```

Scilab code Exa 1.19.a determine whether the signal is energy or power signal

```
// example 1.19<a>
//determine whether the following signals are power
   or energy signal
clc ;
t =0:0.01:100;
A=1;
x=A*sin(t);
P=(integrate('(A*sin(t))^2','t',-%pi,%pi))/(2*%pi);
disp(P,'The power of the signal is:');
E=(integrate('(A*sin(t))^2','t',-%pi,%pi));
disp(E,'The energy is:');
disp('As t tends to infinity energy also tends to
   infinity but power remains finite hence it is
   power signal');
```

Scilab code Exa 1.19.b determine whether the signal is energy or power signal

```
//example 1.19<d>
//determine whether the following signals are power
   or energy signal
clc ;
t=0:0.01:100;
x=1;
P=(integrate('1^2','t',0,1))/2;
disp(P,'The power of the signal is:');
E=(integrate('1^2','t',0,1));
disp(E,'The energy is:');
```

```
disp('As t tends to infinity energy also tends to infinity but power remains finite hence it is power signal');
```

Scilab code Exa 1.19.e determine whether the signal is energy or power signal

```
//example 1.19<e>
//determine whether the following signals are power or energy signal power and energy
clc ;
t=0:0.01:100;
x=t;
T=2;
P=(integrate('t^2','t',0,T))/(T);
disp(P,'The power of the signal is:');
disp('As t tends to infinity energy also tends to infinity but power remains finite hence it is power signal');
```

Scilab code Exa 1.23 find even and odd component of the signal

```
//Example 1.23
//Find the even and odd components of the signal x(t)=(e^-2t)*cos(t)
clc;
clear all;
t=-10:.1:10;
for j=1:length(t)
i=t(j);
x(j)=(exp(-2*i))*cos(i);
y(j)=(exp(2*i))*cos(i);
e(j)=(1/2)*(x(j)+y(j));
o(j)=(1/2)*(x(j)-y(j));
```

```
end
disp('In the plot even component is in red and odd
   component is in blue')
plot(t,e,'red')
plot(t,o,'blue')
```

Scilab code Exa 1.25.a determine whether the system is linear or not

```
//Example 1.25<a>
//To check whether the given discrete system is a
   Linear System (or) Non-Linear System y(t)= t*x(t)
clear;
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('Since It satisifies the superposition principle')
disp('The given system is a Linear system')
else
disp('Since It does not satisify the superposition principle')
disp('The given system is a Non-Linear system')
end
```

Scilab code Exa 1.25.b determine whether the system is linear or not

```
//Example 1.25<b>
//Check whether the following signal is linear or not.
clear ;
close ;
clc ;
T =20; //length of the signal
for n=1: T
x1(n)=n;x2(n)=2*n;
y1(n)=x1(n)*x1(n);
y2(n)=x2(n)*x2(n);
end
z=y1+y2;
for n =1: T
y3(n)=( x1(n)+x2(n)) ^2;
end
if z== y3 then
disp('The following signal is linear');
else
disp ( 'The following signal is non linear');
end
```

Scilab code Exa 1.25.d determine whether the system is linear or not

```
//Example 1.25(d)
//To check whether the given discrete system is a
   Linear System (or) Non-Linear System y[n])= 2*x[n
   ]-3
clear;
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)=2*x1(n)-3;
y2(n)=2*x2(n)-3;
y3(n)=2*x3(n)-3;
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('Since It satisifies the superposition
   principle')
disp('The given system is a Linear system')
else
disp('Since It does not satisify the superposition
   principle')
disp('The given system is a Non-Linear system')
end
```

Scilab code Exa 1.27.a determine whether the system is time invarient or not

```
//Example 1.27<a>
//Determine whether the following system is time
   invariant or not
clc;
clear all;
T =20;
s =2;
for n=1:T
x(n)=n;
y(n)=n*x(n);
end
IP=x(T-s);
OP=y(T-s);
if IP == OP then
disp('The given system is time invariant');
else
disp('The given system is time variant');
end
```

Scilab code Exa 1.29.a determine the stability of system

```
//Example 1.29<a>:
//Determination of stablility of a given system
clear;
clc;
x=[1,2,3,4,0,2,1,3,5,8];
Maximum_Limit=10;
S=0;
for t=0:Maximum_Limit-1
```

```
S=S+t*x(t+1);
end
if (S >Maximum_Limit)
disp('Eventhough input is bounded output is
   unbounded')
disp('The given system is unstable');
else
disp('The given system is stable');
end
```

Scilab code Exa 1.31.d determine the signal is periodic or not

```
//Example 1.31<d>
//check the given signL is PERIODIC OR NOT e^(-2*t)*
   u(t)
clc ;
t=-1:.01:1;
x=exp(-2*t);
plot (t,x);
disp ('(a)this shows that the given signal is not
   periodic which gives w0=1+j,complex no. the
   frequency of signal can never be complex it must
   have real value');
```

Scilab code Exa 1.35.a determine whether the signal is causal or not

```
//Example 1.35<a>
//Find whether the given system is causal or not y(t
   )=x*sin(t).
clear all;
clc;
t=-10:10;
x=2;
```

```
for i=1:length(t)
x3(i)=x*sin(t(i));
end
causal=%t;
for i=1:length (t)
if t(i)<0 & x3(i)~=0 then
noncausal=%f;
end
end
disp (noncausal,"The statement that the system is
    noncausal is");
```

Scilab code Exa 1.36.f determine whether the signal is causal or not

```
//Example 1.36<f>
//Find whether the given signal is causal or not y(n
    )=x(n^2).
clear all;
clc;
n=-10:10;
for i=1:length (n)
x(i)=i;
y(i)=(i.^2) ;
end
causal=%t;
for i=1: length (n)
if n(i)<0 &y(i)~=0 then
causal=%f;
end
end
disp(causal,"The statement that the system is causal
    is:");
```

Chapter 2

convolution and correlation

Scilab code Exa 2.1 `find the convolution of two continuous time signal`

```
//Example 2_1
//Find the convolution of two continous time signal
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 the convolution of two continuous time signal

```
//Example 2_2
//Find the convolution of two continuous time signal
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 2.3 find the convolution of two continuous time signal

```
//Example 2_3
//Find the convolution of two continuous time signal
clc;
t=-8:1/100:8;
for i=1:length (t)
x(i)=exp(-abs(t(i)));
if t(i)>=1 then
h(i)=exp(-2*t(i));
```

```
else
h(i)=0;
end
end
t1=t;
y= convol (x,h)
figure
plot2d(t1,h);
title('Impul seresponce');
figure
plot2d(t,x);
title('Input signal');
figure
t2=-16:1/100:16
plot2d(t2,y);
title('Output signal');
```

Scilab code Exa 2.13 find convolution integral of the signal

```
//Example 2_13
//Convolution Integral of input x(t)=u(t+0.5)-u(t
   -0.5) h(t)=exp(%i*w*t)
clc;
clear;
Max_Limit=0.1;
x=[0,0,0,ones(1,Max_Limit+0.5)-ones(1,Max_Limit-0.5)
   ];
w=2;
t=-9:0;
h=exp(%i*w*t);
N2=0:length(x)-1;
N1=-length(h)+1:0;
t1=-6:3;
y1=(1/%i*w)*(exp(%i*w*(t1+0.5)));
y2=(1/%i*w)*(ones(1,Max_Limit));
```

```
y=[y1 y2]
N=-length(x)+1:length(h)-1;
figure
a=gca();
a.y_location="origin";
plot2d(t,h)
xtitle('Input Response','t','h(t)');
a.thickness=2;
figure
a=gca();
a.y_location="origin";
a.x_location="origin";
a.data_bounds=[-10,0;13,1];
plot2d(-Max_Limit+4:Max_Limit+3,y)
xtitle('Output Response','t','y(t)');
a.thickness=2;
```

Scilab code Exa 2.15 find the convolution of two continuous time signal

```
//Example 2_15
//Find the convolution of two sequences
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 2.16 find the convolution of two continuous time sequences

```
//Example 2_15
//Find the convolution of two sequences
clc;
n=-8:1/1000:8;
for i=1:length(n)
x(i)=exp(-n(i)^2);
if n(i)<0 then
h(i)=exp(n(i));
else
h(i)=exp(-n(i));
end
end
y=convol(x,h);
figure
plot2d3(n,h);
title('Impulse responce');
figure
plot2d3(n,x);
title('Input signal');
figure
n1=-16:1/1000:16
plot2d3(n1,y);
title('Output signal');
```

Scilab code Exa 2.17 find the convolution of two continuous time sequences

```
//Example 2_17
```

```
//find convolution of two sequences
clc;
n=-8:1/1000:8;
for i=1:length(n)
if n(i)<-5 then
x(i)=0;
else
x(i)=(1/2)^n(i);
end
if n(i)<3 then
h(i)=0;
else
h(i)=(1/3)^n(i);
end
end
y=convol(x,h);
figure
plot2d3(n,h);
title('Impulse responce');
figure
plot2d3(n,x);
title('Input signal');
figure
n1=-16:1/1000:16
plot2d3(n1,y);
title('Output signal');
```

Scilab code Exa 2.18 find convolution of rectangular signal

```
//clear//
//Example 2.8:Convolution Integral of input x(t)=(e
    ^2t).u(-t) and
//h(t)=u(t-3)
clear;
close;
```

```
clc;
Max_Limit=0.1;
x=[0,0,0,ones(1,Max_Limit+0.5)-ones(1,Max_Limit-0.5)];
w=2;
t = -9:0;
h= exp(%i*w*t);
N2 = 0:length(x)-1;
N1 = -length(h)+1:0;
t1 = -6:3;
y1 =(1/%i*w)*(exp(%i*w*(t+0.5)));
y2 =(1/%i*w)*(ones(1,Max_Limit));
y = [y1 y2]
N = -length(x)+1:length(h)-1;
figure
a=gca();
a.y_location = "origin";
plot2d(t,h)
xtitle('Input Response','t','h(t)');
a.thickness = 2;
figure
a=gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-10,0;13,1];
plot2d(-Max_Limit+4:Max_Limit+3,y)
xtitle('Output Response','t','y(t)');
a.thickness = 2;
```

Scilab code Exa 2.19.a compute the convolution of the following signals

```
//Example 2_19 <a>
//compute the Convolution of x[n] and Unit Impulse response h[n]
clear;
```

```
close;
clc;
Max_Limit=10;
for n=1:Max_Limit
Alpha=0.5;
h=ones(1,Max_Limit);
N1=0:Max_Limit-1;
x(n)=1;
end
N2=0:Max_Limit-1;
y=convol(x,h);
N=0:2*Max_Limit-2;
figure
a=gca();
plot2d3('gnn',N2,x)
xtitle('Input Response Fig 2.5.(a)','n','x[n]');
a.thickness=2;
figure
a=gca();
plot2d3('gnn',N(1:Max_Limit),y(1:Max_Limit),5)
xtitle('Output Response Fig 2.7','n','y[n]');
a.thickness=2;
```

Scilab code Exa 2.19.b compute the convolution of the following signals

```
//Example 2_19 <b>
//compute the Convolution of x[n] and Unit Impulse
   response h[n]
clear;
close;
clc;
Max_Limit=10;
for n=1:Max_Limit
Alpha=0.5;
h=(0.4)^(n)*ones(1,Max_Limit);
```

```
N1=0:Max_Limit-1;
x(n)=(0.8)^(n-1);
end
N2=0:Max_Limit-1;
y=convol(x,h);
N=0:2*Max_Limit-2;
figure
a=gca();
plot2d3('gnn',N2,x)
xtitle('Input Response Fig 2.5.(a)','n','x[n]');
a.thickness=2;
figure
a=gca();
plot2d3('gnn',N(1:Max_Limit),y(1:Max_Limit),5)
xtitle('Output Response Fig 2.7','n','y[n]');
a.thickness=2;
```

Scilab code Exa 2.19.c compute the convolution of the following signals

```
//Example 2_19 <c>
//compute the Convolution of x[n] and Unit Impulse
   response h[n]
clear;
close;
clc;
Max_Limit=10;
for n=1:Max_Limit
Alpha=0.5;
h=Alpha^(n)*ones(1,Max_Limit);
N1=0:Max_Limit-1;
x(n)=(Alpha^(n-1))*1;
end
N2=0:Max_Limit-1;
y=convol(x,h);
N=0:2*Max_Limit-2;
```

```
figure
a=gca();
plot2d3('gnn',N2,x)
xtitle('Input Response Fig 2.5.(a)','n','x[n]');
a.thickness=2;
figure
a=gca();
plot2d3('gnn',N(1:Max_Limit),y(1:Max_Limit),5)
xtitle('Output Response Fig 2.7','n','y[n]');
a.thickness=2;
```

Scilab code Exa 2.20 determine the step response of the LTI system

```
//Example 2_20
//determine step response of the LTI system
clc;
n=-8:1/1000:8;
for i=1:length(n)
if n(i)>=0 then
x(i)=1;
h(i)=n(i);
else
x(i)=0
h(i)=0;
end
end
y=convol(x,h);
figure
plot2d3(n,h);
title('Impulse responce');
figure
plot2d3(n,x);
title('Input signal');
figure
n1=-16:1/1000:16
```

```
plot2d3(n1,y);
title('Output signal');
```

Chapter 3

continuous time fourier series

Scilab code Exa 3.8 `determine trignom fourier series of half wave rectified sine w`

```
//Example 3_8//
//Fourier Series of half-wave rectifier output
//Assume the period of the signal T=1
t=-0.5:0.01:1;
T=1;
for i=1:length(t)
if t(i)<T/2 then
x(i)=sin(2*%pi*t(i));
else
x(i)=0;
end
end
k=-10:10;
for i=1:length(k)
if k(i)==1 then
ak(i)=1/(4*%i);
elseif k(i)==-1
ak(i)=-1/(4*%i);
else
ak(i)=(cos(k(i)*%pi/2)*exp(-k(i)*%pi/2*-%i))/(%pi-(
    %pi*k(i)*k(i)));
```

```
end
end
disp("The fourier series coefficients are")
disp(ak)
disp("magnitude of Fourier series coefficient")
disp(abs(ak))
//Plotting frequency spectrum
subplot(2,1,1)
plot(k,abs(ak),'.');
xtitle("Magnitude Spectrum","k","jakj");
for i=1:length(k)
if k(i)==0|k(i)==3|k(i)==-3|k(i)==-5|k(i)==5 then
phase(i)=0;
elseif k(i)==-1 then
phase(i)=%pi/2;
elseif k(i)==1 then
phase(i)=-%pi/2;
elseif k(i)== -2|k(i)==-4
phase(i)=%pi;
elseif k(i)==2|k(i)==4
phase(i)=-%pi;
end
end
```

Scilab code Exa 3.11 `find exponential fourier series and sketch corresponding spec`

```
//example 3_11
//exponential fourier Series coefficient and corresponding spectra
clc;
clear;
close;
//Assume period of the impulse train T=2
T=2;
t=-5*T:T:5*T;
```

```
for i=1:length(t)
x(i)=1;
end
//Using shifting property of the impulse signal//
k=-10:10
for i=1:length(k)
ak(i)=1/T;
end
subplot(2,1,1)
plot(t,x,'.')
xtitle("Impulse train","t","x(t)")
subplot(2,1,2)
plot(k,ak,'.')
xtitle("Fourier coefficients of impulse train","k","
   ak")
```

Scilab code Exa 3.13 find fourier series coefficient of periodic signal

```
//Example 3_13
//Continuous Time Fourier Series Coefficients of a
   periodic signal x(t)=2+4*sin((5*%pi)/3*t)+cos((2*
   %pi/3)*t)
clear;
clc;
t=0:0.01:1;
xt=2+4*sin((5*%pi)/3*t)+cos((2*%pi/3)*t);
x_t=2+4*sin((5*%pi)/3*-t)+cos((2*%pi/3)*-t);
x=2+(1/2)*exp(%i*(2*%pi/3)*t)+(1/2)*exp(-%i*(2*%pi
   /3)*t)+(4/(2*%i))*exp(%i*(5*%pi/3)*t)-(4/(2*%i))*
   exp(-%i*(5*%pi/3)*t);
a0=1;
a2=(1/2)
a_2=(1/2)
a3=(4/(2*%i));
a_3=-(4/(2*%i));
```

```
ak=[zeros(1,5) a_3 a_2 0 a2 a3 zeros(1,5)];
k=-7:7;
disp("The fourier series coefficients are...")
disp(ak)
disp("magnitude of Fourier series coefficent")
disp(abs(ak))
subplot(2,1,1)
plot(k,abs(ak),'.');
xtitle("Magnitude Spectrum","k","jakj");
if xt==x_t then
disp("The Given signal is even. It has no phase spectrum");
else
phase=[zeros(1,6) ,%pi/2,0,-%pi/2,zeros(1,6)];
disp("Phase of Fourier series coefficient in radians")
disp(phase)
subplot(2,1,2)
plot(k,phase,'.');
xtitle("Phase Spectrum","k","ak in radians");
end
```

Chapter 4

discrete time fourier series

Scilab code Exa 4.1 determine fourier series coefficient of signal plot its magnit

```
//example 4_1
//determine the fourier series coefficient of the signal plot its magnitude and phase spectrum
clear;
close;
clc;
N = 10;
n = 0:0.01:N;
Wo = 2*%pi/N;
xn =ones(1,length(n))+sin(Wo*n)+3*cos(Wo*n)+cos(2*Wo*n+%pi/2);
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))<=0.1)
    a(k+1)=0;
  end
end
a =a';
a_conj =conj(a);
ak = [a_conj($:-1:1),a(2:$)];
```

```
Mag_ak = abs(ak);
for i = 1:length(a)
  Phase_ak(i) = atan(imag(ak(i))/(real(ak(i))+0.0001));
end
Phase_ak = Phase_ak'
Phase_ak = [Phase_ak(1:$-1) -Phase_ak($:-1:1)];
k = -(N-2):(N-2);
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,real(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('Real part of(ak)')
xlabel('

   k')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,imag(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('imaginary part of(ak)')
xlabel('

   k')
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
```

```
plot2d3('gnn',k,Mag_ak,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('abs(ak)')
xlabel('

   k')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,Phase_ak,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('<(ak)')
xlabel('

   k')
```

Scilab code Exa 4.2 evaluate the fourier series for discrete time periodic square

```
//Example 4_2
//Finad DTFS coefficients of periodic square wave
clc;
clear;
N=10;
N1=2;
Wo=2*%pi/N;
xn=ones(1,length(N));
n=-(2*N1+1):(2*N1+1);
a(1)=(2*N1+1)/N;
for k=1:2*N1
a(k+1)=sin((2*%pi*k*(N1+0.5))/N)/sin(%pi*k/N);
a(k+1)=a(k+1)/N;
if(abs(a(k+1))<=0.1)
```

```
a(k+1)=0;
end
end
a=a';
a_conj=conj(a);
ak=[a_conj($:-1:1),a(2:$)];
k=-2*N1:2*N1;
plot2d3('gnn',k,abs(ak))
xtitle('Magnitude spectrum','k','jakj')
```

Scilab code Exa 4.3 find DTFS coefficient of N periodic impulse train

```
//Example 4_3
//Discrete Time Fourier Transform coefficient of
   Periodic Impulse Train
clc;
clear;
N=5;
N1=-3*N:3*N;
xn=[zeros(1,N-1),1];
x=[1 xn xn xn xn xn xn ];
ak=1/N;
XW=2*%pi*ak*ones(1,2*N);
Wo=2*%pi/N;
n=-N:N-1;
W=Wo*n;
figure
subplot(2,1,1)
plot2d3('gnn',N1,x,2);
xtitle('Periodic Impulse Train','n','x[n]')
subplot(2,1,2)
plot2d3('gnn',W,XW,2);
xtitle('DTFT of Periodic Impulse Train','w','jX(exp(
   jw))j')
disp(Wo)
```

Scilab code Exa 4.12 determine DTFS

```
//Example 4_12_b
clc;
clear;
N=8;
n=0:0.01:N;
Wo=3*%pi/N;
xn=1*ones(1,length(n))+1*sin(Wo*n+%pi/4);
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))<=0.1)
a(k+1)=0;
end
end
a=a';
a_conj=conj(a);
ak=[a_conj($:-1:1),a(2:$)];
Mag_ak=abs(ak);
k=-(N-2):(N-2);
plot2d3('gnn',k,Mag_ak,5)
xtitle('abs(ak)','k','ak')
```

Scilab code Exa 4.14.a determine the fourier series coefficient

```
//Example 4_14_a
//determine the fourier series coeffient of x(n)
clc;
clear;
N=6;
```

```
n=0:0.01:N;
Wo=2*%pi/N;
xn=1*ones(1,length(n))+1*cos(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))<=0.1)
a(k+1)=0;
end
end
a=a';
a_conj=conj(a);
ak=[a_conj($:-1:1),a(2:$)];
Mag_ak=abs(ak);
k=-(N-2):(N-2);
plot2d3('gnn',k,Mag_ak,5)
xtitle('abs(ak)','k','ak')
```

Chapter 5

continuous time fourier transform

Scilab code Exa 5.1 determine continuous time fourier transform of given signal

```
//Example 5.1:
//Continuous Time Fourier Transform of a
clear;
clc;
close;
// Analog Signal
A =1;//Amplitude
Dt = 0.005;
t = 0:Dt:10;
xt=exp(-A*t);
//Continuous-time Fourier Transform
Wmax=2*%pi*1; //Analog Frequency = 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)];
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
a=gca();
a.y_location="origin";
plot(t,xt);
xlabel('t in sec.');
ylabel('x(t)')
title('Continuous Time Signal')
//Plotting Magnitude Response of CTS
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)')
//Plotting Phase Reponse of CTS
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')
```

Scilab code Exa 5.2 determine continuous time fourier transform of given signal

```
//Example 5.2:
//Continuous Time Fourier Transform of x(t)=e-a|t|
clc;
close;
a=1;
Dt=0.005;
```

```
t=-4.5:Dt:4.5;
xt=exp(-a*abs(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=real(XW);
W=[-mtlb_fliplr(W), W(2:1001)];
XW=[mtlb_fliplr(XW),XW(2:1001)];
subplot(1,1,1)
subplot(2,1,1);
a=gca();
a.y_location="origin";
plot(t,xt);
xlabel('t in sec.');
ylabel('x(t)')
title('Continuous Time Signal')
subplot(2,1,2);
a=gca();
a.y_location = "origin";
plot(W,XW);
xlabel('Frequency in Radians/Seconds W');
ylabel('X(jW)')
title('Continuous-time Fourier Transform')
```

Scilab code Exa 5.6 determine fourier transform of signal

```
//Example5_6
//Find Fourier Transform of a Rectangular Waveform x
    (t)=A, from -T0 to T0
clear;
clc;
A=1;
Dt=0.005;
```

```
T0=4;
t=-T0/2:Dt:T0/2;
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);
plot(t,xt);
xlabel('t in sec.');
title('Contiuous Time Signal x(t)')
subplot(2,1,2);
plot(W,XW_Mag );
xlabel('Frequency in Radians/Seconds');
title('Continuous-time Fourier Transform X(jW)')
```

Scilab code Exa 5.8 find the fourier transform

```
//Example 5_8
//find the fourier transform x(t)=exp(-%pi*(t^2))
clear;
clc;
B=%pi;
Dt=0.005;
t=-4.5:Dt:4.5;
xt=exp(-%pi*(t^2));
Wmax=2*%pi*1;
K=4;
```

```
k=0:(K/1000):K;
W=k*Wmax/K;
XW=xt*exp(-sqrt(-1)*t'*W)*Dt;
XW=real(XW);
W=[-mtlb_fliplr(W),W(2:1001)];
XW=[mtlb_fliplr(XW),XW(2:1001)];
subplot(2,1,1);
plot(t,xt);
xlabel('t i n s e c');
ylabel('x(t)')
title('Continuous Time Signal')
subplot(2,1,2);
plot(W,XW);
xlabel('Fr equency in Radians/Seconds W');
ylabel('X(jW)')
title('Continuous-time Fourier Transform')
```

Scilab code Exa 5.9 determine fourier transform of signal

```
//Example 5.9
//Continuous Time Fourier Transforms of cos(Wot)
clc;
close;
//CTFT
T1=2;
T=4*T1;
Wo=2*%pi/T;
W=[-Wo,0,Wo];
ak=(2*%pi*Wo*T1/%pi);
XW1=[-ak,0,ak];
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 5.10 determine fourier transform of signal

```
//Example 5.10
// Continuous Time Fourier Transforms of sin(Wot)
clc;
close;
//CTFT
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];
//figure
a=gca();
a.y_location="origin";
a.x_location="origin";
plot2d3('gnn',W,imag(XW),2);
poly1=a.children(1).children(1);
poly1.thickness = 3;
xlabel('W');
title('CTFT of sin(Wot)')
```

Scilab code Exa 5.15 find the fourier transform

```
//Example 5_15
//Fourier Transform of x(t)=exp(-t)*sin(wc*t)*u(t)
clear;
```

```
clc;
wc=1;
Dt=0.005;
t=0:Dt:10;
xt=(exp(t*(-1+wc))-exp(t*(-1-wc)))/(2*%i);
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)];
//Plotting Continuous Time Signal
figure(1)
plot(t,xt);
xlabel('t in sec.');
ylabel('x(t)')
title('Continuous Time Signal')
figure(2)
//Plotting Magnitude Response of CTS
subplot(2,1,1);
plot(W,XW_Mag);
xlabel('Frequency in Radians/Seconds—>W');
ylabel('abs(X(jW))')
title('Magnitude Response (CTFT)')
//Plotting Phase Reponse of CTS
subplot(2,1,2);
plot(W,XW_Phase*%pi/180);
xlabel('Frequency in Radians/Seconds—>W');
ylabel('<X(jW)')
title('Phase Response (CTFT) in Radians')
```

Scilab code Exa 5.16 find the fourier transform

```
//Example 5_16
//Fourier Transform of x(t)=exp(-a*t)*cos(wc*t)*u(t)
clear;
clc;
a=1;
wc=1;
Dt=0.005;
t=0: Dt :10;
xt=(exp(t*(-a+wc))+exp(t*(-a-wc)))/2;
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)];
//Plotting Continuous Time Signal
figure(1)
plot(t,xt);
xlabel('t in sec.');
ylabel('x(t)')
title('Continuous Time Signal')
figure(2)
//Plotting Magnitude Response of CTS
subplot(2,1,1);
plot(W,XW_Mag);
xlabel('Frequency in Radians/Seconds>W');
ylabel('abs(X(jW))')
title('Magnitude Response (CTFT)')
//Plotting Phase Reponse of CTS
subplot(2,1,2);
plot(W,XW_Phase*%pi/180);
xlabel('Frequency in Radians/Seconds>W');
```

```
ylabel('<X(jW)')
title('Phase Response (CTFT) in Radians')
```

Scilab code Exa 5.17 `find the fourier transform`

```
//Example 5_17
//Fourier Transform of Continuous Time Signal x(t)=cos(wc*t)*u(t)
clear;
clc;
wc=1;
Dt=0.005;
t=0: Dt :10;
xt=(exp(wc*t)+exp(-wc*t))/2;
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)];
//Plotting Continuous Time Signal
figure(1)
plot(t,xt);
xlabel('t in sec.');
ylabel('x(t)')
title('Continuous Time Signal')
figure(2)
//Plotting Magnitude Response of CTS
subplot(2,1,1);
plot(W,XW_Mag);
xlabel('Frequency in Radians/Seconds>W');
```

```
ylabel('abs(X(jW))')
title('Magnitude Response (CTFT)')
//Plotting Phase Reponse of CTS
subplot(2,1,2);
plot(W,XW_Phase*%pi/180);
xlabel('Frequency in Radians/Seconds--->W');
ylabel('<X(jW)')
title('Phase Response (CTFT) in Radians')
```

Scilab code Exa 5.26 `find and sketch fourier transform`

```
//Example 5_26
//find and sketch Fourier Transform of Periodic
   Impulse Train
clear;
clc;
T=-4:4;;
T1=1;
xt=ones(1,length(T));
ak=1/T1;
XW=2*%pi*ak*ones(1,length(T));
Wo=2*%pi/T1;
W=Wo*T;
figure
subplot(2,1,1)
plot2d3('gnn',T,xt);
xlabel('t');
title('Periodic Impulse Train')
subplot(2,1,2)
plot2d3('gnn',W,XW);
xlabel('t');
title('CTFT of Periodic Impulse Train')
```

Chapter 6

discrete time fourier transform

Scilab code Exa 6.2 find fourier transform of causal sequence

```
//Example 6.2:
//Discrete Time Fourier Transform of discrete
   sequence x[n]= (a^n).u[n], a>0 and a<0
clear;
clc;
close;
// DTS Signal
a1 = 0.5;
a2 = -0.5;
max_limit = 10;
for n = 0:max_limit-1
  x1(n+1) = (a1^n);
  x2(n+1) = (a2^n);
end
n = 0:max_limit-1;
Wmax = 2*%pi;
K = 4;
k = 0:(K/1000):K;
W = k*Wmax/K;
x1 = x1';
x2 = x2';
```

```
XW1 = x1* exp(-sqrt(-1)*n'*W);
XW2 = x2* exp(-sqrt(-1)*n'*W);
XW1_Mag = abs(XW1);
XW2_Mag = abs(XW2);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -Wmax to Wmax
XW1_Mag = [mtlb_fliplr(XW1_Mag), XW1_Mag(2:1001)];
XW2_Mag = [mtlb_fliplr(XW2_Mag), XW2_Mag(2:1001)];
[XW1_Phase,db] = phasemag(XW1);
[XW2_Phase,db] = phasemag(XW2);
XW1_Phase = [-mtlb_fliplr(XW1_Phase),XW1_Phase(2:1001)];
XW2_Phase = [-mtlb_fliplr(XW2_Phase),XW2_Phase(2:1001)];
figure
subplot(3,1,1);
plot2d3('gnn',n,x1);
xtitle('Discrete Time Sequence x[n] for a>0')
subplot(3,1,2);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW1_Mag);
title('Magnitude Response abs(X(jW))')
subplot(3,1,3);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW1_Phase);
title('Phase Response <(X(jW))')
//plot for a<0
figure
subplot(3,1,1);
plot2d3('gnn',n,x2);
xtitle('Discrete Time Sequence x[n] for a>0')
subplot(3,1,2);
```

Scilab code Exa 6.5 find fourier transform of noncausal sequence

```
//Example 6.5:
//Find The Fourier Transform
clc;
a = 0.5;
max_limit = 10;
n = -max_limit+1:max_limit-1;
x = a^abs(n);
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)];
XW_Mag = [mtlb_fliplr(XW_Mag), XW_Mag(2:1001)];
//plot for abs(a)<1
figure
subplot(2,1,1);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',n,x);xtitle('Discrete Time Sequence x[
   n] for a>0')
subplot(2,1,2);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW_Mag);
title('Discrete Time Fourier Transform X(exp(jW))')
```

Scilab code Exa 6.6 `find fourier transform of rectangular pulse`

```
//Example 6.6:
//Discrete Time Fourier Transform of
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
figure
subplot(2,1,1);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',n,x);
xtitle('Discrete Time Sequence x[n]')
subplot(2,1,2);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW_Mag);
title('Discrete Time Fourier Transform X(exp(jW))')
```

Scilab code Exa 6.12 `find fourier transform of exponential sequence`

```
//Example 6_12:
//Discrete Time Fourier Transform of discrete
    sequence x[n]=((-1)^n)*(a^n).u[n], a>0 and a<0
clear;
clc;
close;
//DTS Signal
a1=0.5;
a2=-0.5;
max_limit=10;
for n=0:max_limit-1
x1(n+1)=((-1)^n)*(a1^n);
x2(n+1)=((-1)^n)*(a2^n);
end
n=0:max_limit-1;
Wmax=2*%pi;
K=4;
k=0:(K/1000):K;
W=k*Wmax/K;
x1=x1';
x2=x2';
XW1=x1*exp(-sqrt(-1)*n'*W);
XW2=x2*exp(-sqrt(-1)*n'*W);
XW1_Mag=abs(XW1);
XW2_Mag=abs(XW2);
W=[-mtlb_fliplr(W),W(2:1001)];
XW1_Mag = [mtlb_fliplr(XW1_Mag),XW1_Mag(2:1001)];
XW2_Mag = [mtlb_fliplr(XW2_Mag), XW2_Mag(2:1001)];
[XW1_Phase,db] = phasemag(XW1);
[XW2_Phase,db] = phasemag(XW2);
XW1_Phase = [-mtlb_fliplr(XW1_Phase),XW1_Phase
    (2:1001)];
XW2_Phase = [-mtlb_fliplr(XW2_Phase),XW2_Phase
    (2:1001)];
figure
subplot(3,1,1);
plot2d3('gnn',n,x1);
xtitle('Discrete Time Sequence x[n] for a>0')
```

```
subplot(3,1,2);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW1_Mag);
title('Magnitude Response abs(X(jW))')
subplot(3,1,3);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW1_Phase);
title('Phase Response <(X(jW))')
//plot for a<0
figure
subplot(3,1,1);
plot2d3('gnn',n,x2);
xtitle('Discrete Time Sequence x[n] for a>0')
subplot(3,1,2);
```

Scilab code Exa 6.13 determine fourier transform of sequence

```
//Example 6_13:
//Discrete Time Fourier Transform of discrete
   sequence x[n]=(n)*(a^n).u[n], a>0 and a<0
clear;
clc;
close;
//DTS Signal
a1=0.5;
a2=-0.5;
max_limit=10;
for n=0:max_limit-1
x1(n+1)=(n)*(a1^n);
x2(n+1)=(n)*(a2^n);
end
```

```
n=0:max_limit-1;
Wmax=2*%pi;
K=4;
k=0:(K/1000):K;
W=k*Wmax/K;
x1=x1';
x2=x2';
XW1=x1*exp(-sqrt(-1)*n'*W);
XW2=x2*exp(-sqrt(-1)*n'*W);
XW1_Mag=abs(XW1);
XW2_Mag=abs(XW2);
W=[-mtlb_fliplr(W),W(2:1001)];
XW1_Mag = [mtlb_fliplr(XW1_Mag),XW1_Mag(2:1001)];
XW2_Mag = [mtlb_fliplr(XW2_Mag), XW2_Mag(2:1001)];
[XW1_Phase,db] = phasemag(XW1);
[XW2_Phase,db] = phasemag(XW2);
XW1_Phase = [-mtlb_fliplr(XW1_Phase),XW1_Phase(2:1001)];
XW2_Phase = [-mtlb_fliplr(XW2_Phase),XW2_Phase(2:1001)];
figure
subplot(3,1,1);
plot2d3('gnn',n,x1);
xtitle('Discrete Time Sequence x[n] for a>0')
subplot(3,1,2);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW1_Mag);
title('Magnitude Response abs(X(jW))')
subplot(3,1,3);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW1_Phase);
title('Phase Response <(X(jW))')
//plot for a<0
figure
```

```
subplot(3,1,1);
plot2d3('gnn',n,x2);
xtitle('Discrete Time Sequence x[n] for a>0')
subplot(3,1,2);
```

Scilab code Exa 6.18 find and sketch fourier transform of discrete time impulse tr

```
//Example6.18:
//Find and sketch the  Fourier Transform
clc;
N = 5;
N1 = -3*N:3*N;
xn = [zeros(1,N-1),1];
x = [1 xn xn xn xn xn xn];
ak = 1/N;
XW = 2*%pi*ak*ones(1,2*N);
Wo = 2*%pi/N;
n  = -N:N-1;
W = Wo*n;
figure
subplot(2,1,1)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',N1,x,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('n');
title('Periodic Impulse Train')
subplot(2,1,2)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',W,XW,2);
poly1 = a.children(1).children(1);
```

```
poly1.thickness = 3;
xlabel('W');
title('DTFT of Periodic Impulse Train')
disp(Wo)
```

Scilab code Exa 6.19 find fourier transform of periodic signal

```
//Example6.19:
//Discrete Time Fourier Transform
clc;
N = 5;
Wo = 2*%pi/N;
W = [-Wo,0,Wo];
XW =[%pi,0,%pi]; //figure
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',W,XW,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('W');
title('DTFT of cos(nWo)')
disp(Wo)
```

Scilab code Exa 6.20 find the fourier transform of the periodic signal

```
//example 6_20
//find the fourier transform of the periodic signal
    x[n] = cos(Wo)n
clear;
close;
clc;
N = 5;
```

```
n = 0:0.01:N;
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))<=0.1)
    a(k+1)=0;
  end
end
a =a';
a_conj =conj(a);
ak = [a_conj($:-1:1),a(2:$)];
Mag_ak = abs(ak);
k = -(N-2):(N-2);
plot2d3('gnn',k,Mag_ak,5)
xtitle('abs(ak)','k','ak')
```

Chapter 8

sampling

Scilab code Exa 8.1.a determine nyquist rate of given signal

```
//determine the nyquist rate
//8.1(a)
clc;
clear all;
//x(t)=sin(200*pi*t)
wp=200;
F1=wp/2;
Fs=2*F1;
disp('Nyquist Rate=');
disp(Fs);
```

Scilab code Exa 8.1.b determine nyquist rate of given signal

```
//determine the nyquist rate
//8.1(b)
clc;
clear all;
//x(t)=sin2(200*pi*t)
```

```
//x(t)=0.5-0.5cos(400*pi*t)
wp=400;
F1=wp/2;
Fs=2*F1;
disp('Nyquist Rate=');
disp(Fs);
```

Scilab code Exa 8.1.c determine nyquist rate of given signal

```
//determine the nyquist rate
//example 8_1<c>
clc;
clear all;
//x(t)=1+cos(200*pi*t)+sin(400*pi*t)
wq=200;
wp=400;
wf=0;
if wp>=wq then
wf=wp;
else
wf=wq;
end
F1=wf/2;
Fs=2*F1;
disp('Nyquist Rate=');
disp(Fs);
```

Scilab code Exa 8.1.d determine nyquist rate of given signal

```
//determine the nyquist rate
//example 8_1<d>
clc;
clear all;
```

```
//x(t)=cos(150*pi*t)sin(100*pi*t)
//x(t)=0.5sin(250*pi*t)*0.5*sin(50*pi*t)
wq=50;
wp=250;
wf=0;
if wp>=wq then
wf=wp;
else
wf=wq;
end
F1=wf/2;
Fs=2*F1;
disp('Nyquist Rate=');
disp(Fs);
```

Scilab code Exa 8.1.e determine nyquist rate of given signal

```
//determine the nyquist rate
//example 8_1<e>
clc;
clear all;
//x(t)=cos3(200*pi*t)
//cos3(t)=1/4[3cos(t)+cos(3t)]
//cos3(t)=3/4[cos(200)+1/4cos(600)]
wq=600;
wp=200;
wf=0;
if wp>=wq then
wf=wp;
else
wf=wq;
end
F1=wf/2;
Fs=2*F1;
disp('Nyquist Rate=');
```

```
disp(Fs);
```

Scilab code Exa 8.2.a determine nyquist rate of given signal

```
//Example 8_2 <a>
//determine the nyquist rate of x(t)=sinc(200*pi*t)
//sinc(t)=cos(t)/t
//cos3(t)=3/4[cos(200)+1/4cos(600)]
clc;
clear all;
wp=200;
F1=wp/2;
Fs=2*F1;
disp('Nyquist Rate=');
disp(Fs);
```

Scilab code Exa 8.2.b determine nyquist rate of given signal

```
//Example 8_2<b>
//determine the nyquist rate of x(t)=sinc2(200*pi*t)
//sinc(400t)=0.5cos(400t)/400t
clc;
clear all;
wp=400;
F1=wp/2;
Fs=2*F1;
disp('Nyquist Rate=');
disp(Fs);
```

Scilab code Exa 8.2.c determine nyquist rate of given signal

```
//Example 8_2<c>
//determine the nyquist rate of x(t)=sinc(200*pi*t)+sinc2(200*pi*t)
//here,sinc(400t)=0.5cos(400t)/400t+
clc;
clear all;
wq=400;
wp=200;
wf=0;
if wp>=wq then
wf=wp;
else
wf=wq;
end
F1=wf/2;
Fs=2*F1;
disp('Nyquist Rate=');
disp(Fs);
```

Scilab code Exa 8.3.a `determine nyquist criteria satisfy or not`

```
//Example 8_3 <a>
//determine whether the Nyquist criteria satisfy or not
//Ws>=2Wmax
//fs>=2fmax
clc;
clear all;
Ts=0.5*10^-3;
Wc=1000
Fs=1000
Ts_test=1/Fs;
if (Ts<=Ts_test) then
disp('Nyquist Criteria Satify')
else
```

```
disp('Nyquist Criteria NOT Satify ');
end
```

Scilab code Exa 8.3.b determine nyquist criteria satisfy or not

```
//Example 8_3 <b>
//determine whether the Nyquist criteria satisfy or
   not
//Ws>=2Wmax
//fs>=2fmax
clc;
clear all;
Ts=2*10^-3;
Wc=1000
Fs=1000
Ts_test=1/Fs;
if (Ts<=Ts_test) then
disp('Nyquist Criteria Satify')
else
disp('Nyquist Criteria NOT Satify ');
end
```

Scilab code Exa 8.3.c determine nyquist criteria satisfy or not

```
//Example 8_3 <c>
//determine whether the Nyquist criteria satisfy or
   not
//Ws>=2Wmax
//fs>=2fmax
clc;
clear all;
Ts=10^-4;
Wc=1000
```

```
Fs=1000
Ts_test=1/Fs;
if (Ts<=Ts_test) then
disp('Nyquist Criteria Satify')
else
disp('Nyquist Criteria NOT Satify ');
end
```

Scilab code Exa 8.9.a determine minimum sampling frequency

```
//Example 8_9 <a>
//Determine minimum sampling frequency
clc;
clear all;
Fl=9000;
Fh=12000;
Bandwidth_1=Fh-Fl;
a=modulo(Fh,Bandwidth_1);
Fh_1=Fh-a;
div_12=Fh_1./Bandwidth_1;
if(a==0) then
Fs=2*Bandwidth_1;
else
Fs=(2*Fh)/div_12;
end
disp('Minimum Sampling Frequency=');
disp(Fs);
```

Scilab code Exa 8.9.b determine minimum sampling frequency

```
//Example 8_9 <b>
//Determine minimum sampling frequency
clc;
```

```
clear all;
Fl=18000;
Fh=22000;
Bandwidth_1=Fh-Fl;
a=modulo(Fh,Bandwidth_1);
Fh_1=Fh-a;
div_12=Fh_1./Bandwidth_1;
if(a==0) then
Fs=2*Bandwidth_1;
else
Fs=(2*Fh)/div_12;
end
disp('Minimum Sampling Frequency=');
disp(Fs);
```

Scilab code Exa 8.9.c `determine minimum sampling frequency`

```
//Example 8_9 <c>
//Determine minimum sampling frequency
clc;
clear all;
Fl=30000;
Fh=35000;
Bandwidth_1=Fh-Fl;
a=modulo(Fh,Bandwidth_1);
Fh_1=Fh-a;
div_12=Fh_1./Bandwidth_1;
if(a==0) then
Fs=Bandwidth_1;
else
Fs=Bandwidth_1;
end
disp('Minimum Sampling Frequency=');
disp(Fs);
```

Scilab code Exa 8.10.a determine minimum sampling rate

```
//Example 8_10 <a>
//determine the Nyquest rate
//x(t)=10cos(2000)cos(8000)
//x(t)=5cos(6000)+5cos(10000)
clc;
clear all;
wq=10000;
wp=6000;
wf=0;
if wp>=wq then
wf=wp;
else
wf=wq;
end
F1=wf/2;
Fs=2*F1;
disp('Nyquist Rate=');
disp(Fs);
```

Scilab code Exa 8.10.b determine minimum sampling rate

```
//Example 8_10 <b>
//determine the minimum sampling rate
//x(t)=10cos(2000)cos(8000)
//x(t)=5cos(6000)+5cos(10000)
clc;
clear all;
Fl=6000/2;
Fh=10000/2;
Bandwidth_1=Fh-Fl;
```

```
a=modulo(Fh,Bandwidth_1);
Fh_1=Fh-a;
div_12=Fh_1./Bandwidth_1;
if(a==0) then
Fs=2*Bandwidth_1;
else
Fs=(2*Fh)/div_12;
end
disp('Minimum Sampling Frequency=');
disp(Fs);
```

Chapter 10

z transform

Scilab code Exa 10.12.a find z transform

```
//example 10.12(a):
//Find Z transform
clc;
function[za]=ztransfer(sequence,n)
z=poly(0,'z','r')
za=sequence*(1/z)^n'
endfunction
x=[1,2,6,-2,0,3];
n1=0:length(x)-1;
X=ztransfer(x,n1);
disp(X,'X(z)=');
funcprot(0);
```

Scilab code Exa 10.12.b find z transform

```
//example 10.12(b)
//determine Z transform
clc;
```

```
function[za]=ztransfer(sequence,n)
z=poly(0,'z','r')
za=sequence*(1/z)^n'
endfunction
x=[1,2,6,-2,0,3];
n1=-2:length(x)-3;
X=ztransfer(x,n1);
disp(X,'X(z)=');
funcprot(0);
```

Scilab code Exa 10.12.c find z transform

```
//example 10.12(c):
//Find Z transform
clc;
function[za]=ztransfer(sequence,n)
z=poly(0,'z','r')
za=sequence*(1/z)^n'
endfunction
x=[0,0,1,2,6,-2,3];
n1=0:length(x)-1;
X=ztransfer(x,n1);
disp(X,'X(z)=');
funcprot(0);
```

Scilab code Exa 10.12.d find z transform

```
//example 10.12(d)
// Find Z Transform
clc;
function[za]=ztransfer(sequence,n)
z=poly(0,'z','r')
za=sequence*(1/z)^n'
```

```
endfunction
x=[1,2,6,-2,0,3];
n1=-5:length(x)-6;
X=ztransfer(x,n1);
disp(X,'X(z)=');
funcprot(0);
```

Scilab code Exa 10.27 find the convolution of the given signal

```
//Example 10_27
//Convolution of given signals
clc;
x=[1,-2,1];
y=[1,1,1,1,1,1];
X= convol (x,y);
disp(X,'Convolution of given sequences');
```

Scilab code Exa 10.33 find inverse z transform

```
//Example 10_33
//Find the inverse Z-transform
clc;
clear;
z=poly(0,'z');
x=ldiv((z+1),(z-1/3),4);
disp(x,'x[n]=');
```

Scilab code Exa 10.34 determine inverse z transform

```
//Example 10_34
```

```
//Inverse Z–transform using long division method
clc;
clear;
z=poly(0,'z');
x=ldiv(z,(z-0.5),4);
disp(x,'x[n]=');
```

Scilab code Exa 10.41 find the inverse z transform

```
//Example 10_41
//Find the inverse Z–transform using long division
   method
clc;
clear;
z=poly(0,'z');
x=ldiv(z^3-10*z^2-4*z+4,2*z^2-2*z-4,4);
disp(x,'x[n]=');
```

Scilab code Exa 10.53.a determine the unilateral z transform

```
//example 10_53<a>:
//Find unilateral Z transform
clc;
function[za]=ztransfer(sequence,n)
z=poly(0,'z','r')
za=sequence*(1/z)^n'
endfunction
x=[1 2 5 4 0 3];
n1=0:length(x)-1;
disp(n1)
X=ztransfer(x,n1);
disp(X,'X(z)=');
funcprot(0);
```

Scilab code Exa 10.53.b `determine unilateral z transform`

```
//example 10_53<b>:
//Find unilateral Z transform
clc;
function[za]=ztransfer(sequence,n)
z=poly(0,'z','r')
za=sequence*(1/z)^n'
endfunction
x=[1,2,5,4,0,3];
n1=-2:3;
count=0;
for a=n1(1):length(x)
    if a==0 then
        abc=count;
    else
    end
    count=count+1;
end
abc=abc+1;
ac1=0;
x11=[1 1 1 1];
for a=abc:length(x)
    ac1=ac1+1;
    x11(ac1)=x(a);
end
n11=0:(length(x)-abc);
X=ztransfer(x11,n11);
disp(X,'X(z)=');
funcprot(0);
```

Scilab code Exa 10.53.c `determine unilateral z transform`

```
//example 10_53<c>:
//Find unilateral Z transform
clc;
function[za]=ztransfer(sequence,n)
z=poly(0,'z','r')
za=sequence*(1/z)^n'
endfunction
x=[0,0,1,2,5,4,0,3];
n1=0:length(x)-1;
X=ztransfer(x,n1);
disp(X,'X(z)=');
funcprot(0);
```

Chapter 11

state space analysis

Scilab code Exa 11.9 `find state space representation of the system`

```
//Example 11_9
//Find state space representation of the system
clc;
clear;
s=%s;
tf=syslin('c',((3*s+7)/((s+1)*(s+2)*(s+5))));
ss=tf2ss(tf);
disp(ss)
```