

Scilab Textbook Companion for
Signals And Systems
by T. Rawat¹

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

August 23, 2018

¹Funded 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	4
1 introduction to signals and system	5
2 convolution and correlation	25
3 continuous time fourier series	36
4 discrete time fourier series	40
5 continuous time fourier transform	46
6 discrete time fourier transform	56
8 sampling	66
10 z transform	76
11 state space analysis	82

List of Scilab Codes

Exa 1.2.a	sketch the signal	5
Exa 1.2.b	sketch the signal	6
Exa 1.2.c	sketch the signal	6
Exa 1.6.a	draw the waveforms of the siganals	7
Exa 1.6.b	draw the waveforms of the siganals	7
Exa 1.6.c	draw the waveforms of the siganals	8
Exa 1.6.d	draw the waveforms of the siganals	8
Exa 1.6.e	draw the waveforms of the siganals	9
Exa 1.6.f	draw the waveforms of the siganals	9
Exa 1.6.g	draw the waveforms of the siganals	10
Exa 1.7.a	draw the waveforms of the siganals	10
Exa 1.7.b	draw the waveforms of the siganals	11
Exa 1.7.c	draw the waveforms of the siganals	11
Exa 1.7.d	draw the waveforms of the siganals	12
Exa 1.7.e	draw the waveforms of the siganals	13
Exa 1.7.f	draw the waveforms of the siganals	13
Exa 1.8.a	draw the waveforms of the siganals	14
Exa 1.14.a	draw the waveforms of the siganals	14
Exa 1.14.b	determine whether the signal is periodic or not	15
Exa 1.14.c	determine whether the signal is periodic or not	15
Exa 1.15.a	determine whether the signal is periodic or not	15
Exa 1.18.c	determine values of power and energy	16
Exa 1.18.d	determine values of power and energy	16
Exa 1.19.a	determine whether the signal is energy or power signal	17
Exa 1.19.b	determine whether the signal is energy or power signal	17

Exa 1.19.e	determine whether the signal is energy or power signal	18
Exa 1.23	find even and odd component of the signal	18
Exa 1.25.a	determine whether the system is linear or not	19
Exa 1.25.b	determine whether the system is linear or not	20
Exa 1.25.d	determine whether the system is linear or not	21
Exa 1.27.a	determine whether the system is time invariant or not	22
Exa 1.29.a	determine the stability of system	22
Exa 1.31.d	determine the signal is periodic or not	23
Exa 1.35.a	determine whether the signal is causal or not	23
Exa 1.36.f	determine whether the signal is causal or not	24
Exa 2.1	find the convolution of two continuous time signal	25
Exa 2.2	find the convolution of two continuous time signal	26
Exa 2.3	find the convolution of two continuous time signal	26
Exa 2.13	find convolution integral of the signal	27
Exa 2.15	find the convolution of two continuous time signal	28
Exa 2.16	find the convolution of two continuous time sequences	29
Exa 2.17	find the convolution of two continuous time sequences	29
Exa 2.18	find convolution of rectangular signal	30
Exa 2.19.a	compute the convolution of the following signals	31
Exa 2.19.b	compute the convolution of the following signals	32
Exa 2.19.c	compute the convolution of the following signals	33
Exa 2.20	determine the step response of the LTI system	34
Exa 3.8	determine trigonometric fourier series of half wave rectified sine wave	36
Exa 3.11	find exponential fourier series and sketch corresponding spectra	37
Exa 3.13	find fourier series coefficient of periodic signal	38

Exa 4.1	determine fourier series coefficient of signal plot its magnitude and phase spectrum . . .	40
Exa 4.2	evaluate the fourier series for discrete time periodic square wave	42
Exa 4.3	find DTFS coefficient of N periodic impulse train	43
Exa 4.12	determine DTFS	44
Exa 4.14.a	determine the fourier series coefficient	44
Exa 5.1	determine continuous time fourier transform of given signal	46
Exa 5.2	determine continuous time fourier transform of given signal	47
Exa 5.6	determine fourier transform of signal	48
Exa 5.8	find the fourier transform	49
Exa 5.9	determine fourier transform of signal	50
Exa 5.10	determine fourier transform of signal	51
Exa 5.15	find the fourier transform	51
Exa 5.16	find the fourier transform	52
Exa 5.17	find the fourier transform	54
Exa 5.26	find and sketch fourier transform	55
Exa 6.2	find fourier transform of causal sequence . .	56
Exa 6.5	find fourier transform of noncausal sequence	58
Exa 6.6	find fourier transform of rectangular pulse .	59
Exa 6.12	find fourier transform of exponential sequence	59
Exa 6.13	determine fourier transform of sequence . .	61
Exa 6.18	find and sketch fourier transform of discrete time impulse train	63
Exa 6.19	find fourier transform of periodic signal . . .	64
Exa 6.20	find the fourier transform of the periodic signal	64
Exa 8.1.a	determine nyquist rate of given signal	66
Exa 8.1.b	determine nyquist rate of given signal	66
Exa 8.1.c	determine nyquist rate of given signal	67
Exa 8.1.d	determine nyquist rate of given signal	67
Exa 8.1.e	determine nyquist rate of given signal	68
Exa 8.2.a	determine nyquist rate of given signal	69
Exa 8.2.b	determine nyquist rate of given signal	69
Exa 8.2.c	determine nyquist rate of given signal	69
Exa 8.3.a	determine nyquist criteria satisfy or not . .	70

Exa 8.3.b	determine nyquist criteria satisfy or not . . .	71
Exa 8.3.c	determine nyquist criteria satisfy or not . . .	71
Exa 8.9.a	determine minimum sampling frequency . . .	72
Exa 8.9.b	determine minimum sampling frequency . . .	72
Exa 8.9.c	determine minimum sampling frequency . . .	73
Exa 8.10.a	determine minimum sampling rate	74
Exa 8.10.b	determine minimum sampling rate	74
Exa 10.12.a	find z transform	76
Exa 10.12.b	find z transform	76
Exa 10.12.c	find z transform	77
Exa 10.12.d	find z transform	77
Exa 10.27	find the convolution of the given signal . . .	78
Exa 10.33	find inverse z transform	78
Exa 10.34	determine inverse z transform	78
Exa 10.41	find the inverse z transform	79
Exa 10.53.a	determine the unilateral z transform	79
Exa 10.53.b	determine unilateral z transform	80
Exa 10.53.c	determine unilateral z transform	80
Exa 11.9	find state space representation of the system	82

Chapter 1

introduction to signals and system

Scilab code Exa 1.2.a sketch the signal

```
1 //example 1_2<a>
2 //sketch the following signal x(3t)
3 clc;
4 clear all;
5 t=-1/3:0.0001:1/3;
6 for i=1:length(t)
7 if t(i)<0 then
8 x(i)=1+3*t(i);
9 else
10 x(i)=1-3*t(i);
11 end
12 end
13 plot2d(t,x)
14 plot (t,x, 'red' );
15 xtitle('required figure ', 't', 'x(3*t)');
16 xgrid();
```

Scilab code Exa 1.2.b sketch the signal

```
1 //example 1_2<b>
2 //sketch the following signal  $x(3t+2)$ 
3 clc;
4 clear all;
5 t=-1:0.0001:-1/3;
6 for i=1:length(t)
7 if t(i)<-2/3 then
8 x(i)=3+3*t(i);
9 else
10 x(i)=-1-3*t(i);
11 end
12 end
13 plot(t,x)
14 plot(t,x, 'red' );
15 xtitle('required figure ', 't', 'x(3*t+2)');
16 xgrid();
```

Scilab code Exa 1.2.c sketch the signal

```
1 //example 1_2<c>
2 //sketch the following signal  $x(-2t-1)$ 
3 clc;
4 clear all;
5 t=-1:0.0001:0;
6 for i=1:length(t)
7 if t(i)>=-1/2 then
8 x(i)=-2*t(i);
9 else
10 x(i)=(2*t(i))+2;
11 end
12 end
13 plot(t,x)
14 plot(t,x, 'red' );
```

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

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

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

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

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

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

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

```
1 //draw the waveform of the signal  $x_3(t)=u(-t)$ 
2 clc ;
3 clear all;
4 t=-10:.001:10;
5 for i=1:length(t)
6 if t(i)<=0 then
7 x(i)=1;
8 else
9 x(i)=0;
10 end
11 end
12 plot2d(t,x);
13 xtitle('Required figure ','t','x(t)')
```

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

```
1 //draw the waveform of the signal  $x_4(t)=u(-2t+1)$ 
2 clc ;
3 clear all;
4 t=-10:.001:10;
5 for i=1:length(t)
6 if t(i)<=1/2 then
7 x(i)=1;
8 else
9 x(i)=0;
10 end
11 end
```

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

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

```
1 clc ;
2 clear all;
3 t=-10:.001:10;
4 for i=1:length(t)
5 if t(i)<=-1/2 then
6 x(i)=1;
7 else
8 x(i)=0;
9 end
10 end
11 // f i g u r e
12 f=scf(0);
13 plot2d(t,x);
14 xtitle('Required figure','t','x(t)')
```

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

```
1 //Example 1.6<f>
2 //draw the waveform of the signal  $x_6(t)=u(t+2)-u(t-2)$ 
3 clc ;
4 clear all;
5 t=-10:.001:10;
6 for i=1:length(t)
7 if t(i)>=-2 & t(i)<=2 then
8 x(i)=1;
9 else
10 x(i)=0;
```

```

11 end
12 end
13 //figure
14 plot2d(t,x);
15 xtitle('Required figure ','t','x(t)')

```

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

```

1 //Example 1.6<g>//
2 //draw the waveform of the signal  $x_7(t)=u(t)-2*u(t-1)+u(t-2)$ //
3 clc ;
4 clear all;
5 t=-10:0.001:10
6 for i=1:length(t)
7 if t(i)>=0 & t(i)<=1 then
8 x(i)=1;
9 else if t(i)<1 & t(i)>2 then
10 x(i)=0
11 end
12 end
13 if t(i)>1 & t(i)<=2 then
14 x(i)=-1
15 end
16 end
17 //figure
18 plot2d(t,x);
19 xtitle('Required figure ','t','x(t)')

```

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

```

1 //Example 1.7<a>
2 //draw the waveform of the signal  $x_1(t)=r(t-1)$ 

```

```

3  clc;
4  clear all;
5  t=-10:.001:10;
6  for i=1:length(t)
7  if t(i)>=0 then
8  x(i)=t(i)+1;
9  else
10 x(i)=0;
11 end
12 end
13 plot2d(x,t);
14 xtitle('Required figure ','t','x(t)')

```

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

```

1 //Example 1.7<b>
2 //draw the waveform of the signal  $x_1(t)=r(t+1)$ 
3 clc;
4 clear all;
5 t=-10:.001:10;
6 for i=1:length(t)
7 if t(i)>=-1 then
8 x(i)=t(i)+1;
9 else
10 x(i)=0;
11 end
12 end
13 plot2d(t,x);
14 xtitle('Required figure ','x(t)','t')

```

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

```

1 //Example 1.7<c>

```

```

2 //draw the waveform of the signal  $x_1(t)=r(-t)$ 
3 clc;
4 clear all;
5  $t=-10:0.001:10$ ;
6 for  $i=1:\text{length}(t)$ 
7 if  $t(i)\geq 0$  then
8  $x(i)=-t(i)$ ;
9 else
10  $x(i)=0$ ;
11 end
12 end
13 //figure
14 plot2d( $x,t$ );
15 xtitle('Required figure ','t','x(t)')

```

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

```

1 //Example 1.7<d>
2 //draw the waveform of the signal  $x_1(t)=r(3t)$ 
3 clc;
4 clear all;
5  $t=-10:.001:10$ ;
6 for  $i=1:\text{length}(t)$ 
7 if  $t(i)\geq 0$  then
8  $x(i)=3*t(i)$ ;
9 else
10  $x(i)=0$ ;
11 end
12 end
13 //figure
14 plot2d( $x,t$ );
15 xtitle('Required figure ','t','x(t)')

```

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

```
1 //Example 1.7<e>
2 //draw the waveform of the signal  $x_1(t)=r(-3t)$ 
3 clc;
4 clear all;
5 t=-10:.001:10;
6 for i=1:length(t)
7 if t(i)>=0 then
8 x(i)=-3*t(i);
9 else
10 x(i)=0;
11 end
12 end
13 //figure
14 plot2d(x,t);
15 xtitle('Required figure ','t','x(t)')
```

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

```
1 //Example 1.7<f>
2 //draw the waveform of the signal  $x_1(t)=r(3t+1)$ 
3 clc;
4 clear all;
5 t=-10:.001:10;
6 for i=1:length(t)
7 if t(i)>=-1/3 then
8 x(i)=3*t(i)-1/3;
9 else
10 x(i)=0;
11 end
12 end
13 //figure
14 plot2d(x,t);
15 xtitle('Required figure ','t','x(t)')
```

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

```
1 //Example 1.8<a>
2 //draw the waveform of the signal  $x_1(t)=r(t)-r(t-1)-$ 
    $u(t-1)$ 
3 clc;
4 clear all;
5 t=-10:0.001:10;
6 for i=1:length(t)
7 if t(i)>=0 & t(i)<1 then
8 x(i)=t(i);
9 else
10 x(i)=0;
11 end
12 end
13 plot2d(t,x)
14 xtitle('Required figure ','t','x(t)')
```

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

```
1 //Example 1.14<a>
2 //check the given signal is PERIODIC OR NOT//
3 clc ;
4 t=-10:.01:10;
5 x=%i*(exp(%i*10*t));
6 subplot (311)
7 plot (t,x);
8 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

```
1 //Example 1.14<b>
2 //check the given signal is PERIODIC OR NOT  $e^{(-1+j)*t}$ 
3 clc ;
4 t=-1:.01:1;
5 x=(exp(-1+%i)*t);
6 f=1/2*%pi;
7 N=1/f;
8 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

```
1 //Example 1.14<c>
2 // Find whether the following signal is periodic or
   not  $x_3(n)=e^{(i*7*pi*n)}$ 
3 clc;
4 n=-21:21;
5 x=exp(%i *7* %pi *n);
6 f=(7*%pi)/(2*%pi);
7 N=1/f;
8 disp(N,'the given signal is periodic');
```

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

```
1 //Example 1.15<a>
2 // Find whether the following signal is periodic or
   not  $x_3(n)=2*e^{(i*(t+%pi/4))}$ 
3 clc;
4 t=-21:21;
5 x=2*exp(%i*(t+%pi/4));
```

```

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

```

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

```

1 // example 1.18<c>
2 //determine the values of power and energy
3 clc ;
4 t =0:0.01:100;
5 A=1;
6 x=A*cos (t);
7 P=(integrate('(A*cos(t))^2','t',0,2*%pi ))/(2*%pi);
8 disp(P,'The power of the signal is:');
9 E=(integrate('(A*cos(t))^2','t',0,2*%pi));
10 disp(E,'The energy is:');
11 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

```

1 //Example 1.18<d>
2 //determine the values of power and energy
3 clc ;
4 E=0;
5 for n=0:200
6 x(n+1)=(1/2)^n;
7 end
8 for n=0:200
9 E=E+x(n +1)^2;
10 end

```

```

11 disp(E, 'The energy of the signal is; ');
12 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

```

1 // example 1.19<a>
2 //determine whether the following signals are power
  or energy signal
3 clc ;
4 t =0:0.01:100;
5 A=1;
6 x=A*sin(t);
7 P=(integrate('(A*sin(t))^2','t',-%pi,%pi))/(2*%pi);
8 disp(P, 'The power of the signal is:');
9 E=(integrate('(A*sin(t))^2','t',-%pi,%pi));
10 disp(E, 'The energy is:');
11 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

```

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

```

```
10 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

```
1 //example 1.19<e>
2 //determine whether the following signals are power
    or energy signal power and energy
3 clc ;
4 t=0:0.01:100;
5 x=t;
6 T=2;
7 P=(integrate('t^2','t',0,T))/(T);
8 disp(P,'The power of the signal is:');
9 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

```
1 //Example 1.23
2 //Find the even and odd components of the signal  $x(t)$ 
     $= (e^{-2t}) * \cos(t)$ 
3 clc;
4 clear all;
5 t=-10:.1:10;
6 for j=1:length(t)
7 i=t(j);
8 x(j)=(exp(-2*i))*cos(i);
9 y(j)=(exp(2*i))*cos(i);
10 e(j)=(1/2)*(x(j)+y(j));
11 o(j)=(1/2)*(x(j)-y(j));
```

```

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

```

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

```

1 //Example 1.25<a>
2 //To check whether the given discrete system is a
  Linear System (or) Non-Linear System  $y(t)= t*x(t)$ 
3 clear;
4 clc;
5 x1=[1,1,1,1];
6 x2=[2,2,2,2];
7 a=1;
8 b=1;
9 for t=1:length(x1)
10 x3(t)=a*x1(t)+b*x2(t);
11 end
12 for t=1:length(x1)
13 y1(t)=t*x1(t);
14 y2(t)=t*x2(t);
15 y3(t)=t*x3(t);
16 end
17 for t=1:length(y1)
18 z(t)=a*y1(t)+b*y2(t);
19 end
20 count=0;
21 for n=1:length(y1)
22 if(y3(t)==z(t))
23 count=count+1;
24 end
25 end
26 if(count==length(y3))

```

```

27 disp('Since It satisfies the superposition
      principle')
28 disp('The given system is a Linear system')
29 else
30 disp('Since It does not satisfy the superposition
      principle')
31 disp('The given system is a Non-Linear system')
32 end

```

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

```

1 //Example 1.25<b>
2 //Check whether the following signal is linear or
  not.
3 clear ;
4 close ;
5 clc ;
6 T =20; //length of the signal
7 for n=1: T
8 x1(n)=n;x2(n)=2*n;
9 y1(n)=x1(n)*x1(n);
10 y2(n)=x2(n)*x2(n);
11 end
12 z=y1+y2;
13 for n =1: T
14 y3(n)=( x1(n)+x2(n)) ^2;
15 end
16 if z== y3 then
17 disp('The following signal is linear');
18 else
19 disp ( 'The following signal is non linear');
20 end

```

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

```
1 //Example 1.25(d)
2 //To check whether the given discrete system is a
   Linear System (or) Non-Linear System  $y[n]= 2*x[n]$ 
   ]-3
3 clear;
4 clc;
5 x1=[1,1,1,1];
6 x2=[2,2,2,2];
7 a=1;
8 b=1;
9 for n=1:length(x1)
10 x3(n)=a*x1(n)+b*x2(n);
11 end
12 for n=1:length(x1)
13 y1(n)=2*x1(n)-3;
14 y2(n)=2*x2(n)-3;
15 y3(n)=2*x3(n)-3;
16 end
17 for n=1:length(y1)
18 z(n)=a*y1(n)+b*y2(n);
19 end
20 count=0;
21 for n=1:length(y1)
22 if(y3(n)==z(n))
23 count=count+1;
24 end
25 end
26 if(count==length(y3))
27 disp('Since It satisfies the superposition
   principle')
28 disp('The given system is a Linear system')
29 else
30 disp('Since It does not satisfy the superposition
   principle')
31 disp('The given system is a Non-Linear system')
32 end
```

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

```
1 //Example 1.27<a>
2 //Determine whether the following system is time
   invariant or not
3 clc;
4 clear all;
5 T =20;
6 s =2;
7 for n=1:T
8 x(n)=n;
9 y(n)=n*x(n);
10 end
11 IP=x(T-s);
12 OP=y(T-s);
13 if IP == OP then
14 disp('The given system is time invariant');
15 else
16 disp('The given system is time variant');
17 end
```

Scilab code Exa 1.29.a determine the stability of system

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

```

9 S=S+t*x(t+1);
10 end
11 if (S >Maximum_Limit)
12 disp('Eventhough input is bounded output is
        unbounded')
13 disp('The given system is unstable');
14 else
15 disp('The given system is stable');
16 end

```

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

```

1 //Example 1.31<d>
2 //check the given signL is PERIODIC OR NOT  $e^{-2*t}$ *
   u(t)
3 clc ;
4 t=-1:.01:1;
5 x=exp(-2*t);
6 plot (t,x);
7 disp ('(a)this shows that the given signal is not
        periodic which gives  $w_0=1+j$ , 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

```

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

```

```

7 for i=1:length(t)
8 x3(i)=x*sin(t(i));
9 end
10 causal=%t;
11 for i=1:length (t)
12 if t(i)<0 & x3(i)~=0 then
13 noncausal=%f;
14 end
15 end
16 disp (noncausal,"The statement that the system is
noncausal is");

```

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

```

1 //Example 1.36<f>
2 //Find whether the given signal is causal or not y(n
) = x(n^2).
3 clear all;
4 clc;
5 n=-10:10;
6 for i=1:length (n)
7 x(i)=i;
8 y(i)=(i.^2) ;
9 end
10 causal=%t;
11 for i=1: length (n)
12 if n(i)<0 &y(i)~=0 then
13 causal=%f;
14 end
15 end
16 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

```
1 //Example 2_1
2 //Find the convolution of two continous time signal
3 clc;
4 t=-8:1/100:8;
5 for i=1:length(t)
6 x(i)=exp(-t(i)^2);
7 h(i)=3*t(i)^2;
8 end
9 y=convol(x,h);
10 figure
11 plot2d(t,h);
12 title('Impulse responce');
13 figure
14 plot2d(t,x);
15 title('Input signal');
16 figure
17 t2=-16:1/100:16
18 plot2d(t2,y);
19 title('Output signal');
```

Scilab code Exa 2.2 find the convolution of two continuous time signal

```
1 //Example 2_2
2 //Find the convolution of two continuous time signal
3 clc;
4 t=-8:1/100:8;
5 for i=1:length(t)
6 x(i)=3*cos(2.*t(i));
7 h(i)=exp(-abs(t(i)));
8 end
9 y=convol(x,h);
10 figure
11 plot2d(t,h);
12 title('Impulse responce');
13 figure
14 plot2d(t,x);
15 title('Input signal');
16 figure
17 t2=-16:1/100:16
18 plot2d(t2,y);
19 title('Output signal');
```

Scilab code Exa 2.3 find the convolution of two continuous time signal

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

```

9  else
10 h(i)=0;
11 end
12 end
13 t1=t;
14 y= convol (x,h)
15 figure
16 plot2d(t1,h);
17 title('Impul seresponce ');
18 figure
19 plot2d(t,x);
20 title('Input signal ');
21 figure
22 t2=-16:1/100:16
23 plot2d(t2,y);
24 title('Output signal ');

```

Scilab code Exa 2.13 find convolution integral of the signal

```

1 //Example 2_13
2 //Convolution Integral of input  $x(t)=u(t+0.5)-u(t-0.5)$   $h(t)=\exp(i\omega t)$ 
3 clc;
4 clear;
5 Max_Limit=0.1;
6 x=[0,0,0,ones(1,Max_Limit+0.5)-ones(1,Max_Limit-0.5)
    ];
7 w=2;
8 t=-9:0;
9 h=exp(i*w*t);
10 N2=0:length(x)-1;
11 N1=-length(h)+1:0;
12 t1=-6:3;
13 y1=(1/i*w)*(exp(i*w*(t1+0.5)));
14 y2=(1/i*w)*(ones(1,Max_Limit));

```

```

15 y=[y1 y2]
16 N=-length(x)+1:length(h)-1;
17 figure
18 a=gca();
19 a.y_location="origin";
20 plot2d(t,h)
21 xtitle('Input Response','t','h(t)');
22 a.thickness=2;
23 figure
24 a=gca();
25 a.y_location="origin";
26 a.x_location="origin";
27 a.data_bounds=[-10,0;13,1];
28 plot2d(-Max_Limit+4:Max_Limit+3,y)
29 xtitle('Output Response','t','y(t)');
30 a.thickness=2;

```

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

```

1 //Example 2_15
2 //Find the convolution of two sequences
3 clc
4 n=-8:1:8;
5 for i=1:length(n)
6 x(i)=exp(-n(i)^2);
7 h(i)=3.*n(i)^2;
8 end
9 y=convol(x,h);
10 figure
11 plot2d3(n,h);
12 title('Impulse response');
13 figure
14 plot2d3(n,x);
15 title('Input signal');
16 figure

```



```
17 n1=-16:1:16
18 plot2d3(n1,y);
19 title('Output signal');
```

Scilab code Exa 2.16 find the convolution of two continuous time sequences

```
1 //Example 2_15
2 //Find the convolution of two sequences
3 clc;
4 n=-8:1/1000:8;
5 for i=1:length(n)
6 x(i)=exp(-n(i)^2);
7 if n(i)<0 then
8 h(i)=exp(n(i));
9 else
10 h(i)=exp(-n(i));
11 end
12 end
13 y=convol(x,h);
14 figure
15 plot2d3(n,h);
16 title('Impulse response');
17 figure
18 plot2d3(n,x);
19 title('Input signal');
20 figure
21 n1=-16:1/1000:16
22 plot2d3(n1,y);
23 title('Output signal');
```

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

```
1 //Example 2_17
```

```

2 //find convolution of two sequences
3 clc;
4 n=-8:1/1000:8;
5 for i=1:length(n)
6 if n(i)<-5 then
7 x(i)=0;
8 else
9 x(i)=(1/2)^n(i);
10 end
11 if n(i)<3 then
12 h(i)=0;
13 else
14 h(i)=(1/3)^n(i);
15 end
16 end
17 y=convol(x,h);
18 figure
19 plot2d3(n,h);
20 title('Impulse responce');
21 figure
22 plot2d3(n,x);
23 title('Input signal');
24 figure
25 n1=-16:1/1000:16
26 plot2d3(n1,y);
27 title('Output signal');

```

Scilab code Exa 2.18 find convolution of rectangular signal

```

1 //clear//
2 //Example 2.8: Convolution Integral of input  $x(t)=(e^{2t}).u(-t)$  and
    $h(t)=u(t-3)$ 
3 //h(t)=u(t-3)
4 clear;
5 close;

```

```

6  clc;
7  Max_Limit=0.1;
8  x=[0,0,0,ones(1,Max_Limit+0.5)-ones(1,Max_Limit-0.5)
    ];
9  w=2;
10 t = -9:0;
11 h= exp(%i*w*t);
12 N2 = 0:length(x)-1;
13 N1 = -length(h)+1:0;
14 t1 = -6:3;
15 y1 =(1/%i*w)*(exp(%i*w*(t+0.5)));
16 y2 =(1/%i*w)*(ones(1,Max_Limit));
17 y = [y1 y2]
18 N = -length(x)+1:length(h)-1;
19 figure
20 a=gca();
21 a.y_location = "origin";
22 plot2d(t,h)
23 xtitle('Input Response','t','h(t)');
24 a.thickness = 2;
25 figure
26 a=gca();
27 a.y_location = "origin";
28 a.x_location = "origin";
29 a.data_bounds=[-10,0;13,1];
30 plot2d(-Max_Limit+4:Max_Limit+3,y)
31 xtitle('Output Response','t','y(t)');
32 a.thickness = 2;

```

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

```

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

```

```

4  close;
5  clc;
6  Max_Limit=10;
7  for n=1:Max_Limit
8  Alpha=0.5;
9  h=ones(1,Max_Limit);
10 N1=0:Max_Limit-1;
11 x(n)=1;
12 end
13 N2=0:Max_Limit-1;
14 y=convol(x,h);
15 N=0:2*Max_Limit-2;
16 figure
17 a=gca();
18 plot2d3('gnn',N2,x)
19 xtitle('Input Response Fig 2.5.(a)', 'n', 'x[n]');
20 a.thickness=2;
21 figure
22 a=gca();
23 plot2d3('gnn',N(1:Max_Limit),y(1:Max_Limit),5)
24 xtitle('Output Response Fig 2.7', 'n', 'y[n]');
25 a.thickness=2;

```

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

```

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

```

```

10 N1=0:Max_Limit-1;
11 x(n)=(0.8)^(n-1);
12 end
13 N2=0:Max_Limit-1;
14 y=convol(x,h);
15 N=0:2*Max_Limit-2;
16 figure
17 a=gca();
18 plot2d3('ggn',N2,x)
19 xtitle('Input Response Fig 2.5.(a)', 'n', 'x[n]');
20 a.thickness=2;
21 figure
22 a=gca();
23 plot2d3('ggn',N(1:Max_Limit),y(1:Max_Limit),5)
24 xtitle('Output Response Fig 2.7 ', 'n', 'y[n]');
25 a.thickness=2;

```

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

```

1 //Example 2_19 <c>
2 //compute the Convolution of x[n] and Unit Impulse
   response h[n]
3 clear;
4 close;
5 clc;
6 Max_Limit=10;
7 for n=1:Max_Limit
8 Alpha=0.5;
9 h=Alpha^(n)*ones(1,Max_Limit);
10 N1=0:Max_Limit-1;
11 x(n)=(Alpha^(n-1))*1;
12 end
13 N2=0:Max_Limit-1;
14 y=convol(x,h);
15 N=0:2*Max_Limit-2;

```

```

16 figure
17 a=gca();
18 plot2d3('gmn',N2,x)
19 xtitle('Input Response Fig 2.5.(a)', 'n', 'x[n]');
20 a.thickness=2;
21 figure
22 a=gca();
23 plot2d3('gmn',N(1:Max_Limit),y(1:Max_Limit),5)
24 xtitle('Output Response Fig 2.7', 'n', 'y[n]');
25 a.thickness=2;

```

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

```

1 //Example 2_20
2 //determine step response of the LTI system
3 clc;
4 n=-8:1/1000:8;
5 for i=1:length(n)
6 if n(i)>=0 then
7 x(i)=1;
8 h(i)=n(i);
9 else
10 x(i)=0
11 h(i)=0;
12 end
13 end
14 y=convol(x,h);
15 figure
16 plot2d3(n,h);
17 title('Impulse response');
18 figure
19 plot2d3(n,x);
20 title('Input signal');
21 figure
22 n1=-16:1/1000:16

```

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

Chapter 3

continuous time fourier series

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

```
1 //Example 3_8//
2 //Fourier Series of half-wave rectifier output
3 //Assume the period of the signal T=1
4 t=-0.5:0.01:1;
5 T=1;
6 for i=1:length(t)
7 if t(i)<T/2 then
8 x(i)=sin(2*%pi*t(i));
9 else
10 x(i)=0;
11 end
12 end
13 k=-10:10;
14 for i=1:length(k)
15 if k(i)==1 then
16 ak(i)=1/(4*%i);
17 elseif k(i)==-1
18 ak(i)=-1/(4*%i);
19 else
20 ak(i)=(cos(k(i)*%pi/2)*exp(-k(i)*%pi/2*-%i))/(%pi-(
    %pi*k(i)*k(i)));
```



```

21 end
22 end
23 disp("The fourier series coefficients are")
24 disp(ak)
25 disp("magnitude of Fourier series coefficient")
26 disp(abs(ak))
27 //Plotting frequency spectrum
28 subplot(2,1,1)
29 plot(k,abs(ak),'.');
30 xtitle("Magnitude Spectrum","k","jakj");
31 for i=1:length(k)
32 if k(i)==0|k(i)==3|k(i)==-3|k(i)==-5|k(i)==5 then
33 phase(i)=0;
34 elseif k(i)==-1 then
35 phase(i)=%pi/2;
36 elseif k(i)==1 then
37 phase(i)=-%pi/2;
38 elseif k(i)== -2|k(i)==-4
39 phase(i)=%pi;
40 elseif k(i)==2|k(i)==4
41 phase(i)=-%pi;
42 end
43 end

```

Scilab code Exa 3.11 find exponential fourier series and sketch corresponding spectra

```

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

```

```

 9  for i=1:length(t)
10  x(i)=1;
11  end
12  //Using shifting property of the impulse signal//
13  k=-10:10
14  for i=1:length(k)
15  ak(i)=1/T;
16  end
17  subplot(2,1,1)
18  plot(t,x, ' . ')
19  xtitle("Impulse train", "t", "x(t)")
20  subplot(2,1,2)
21  plot(k,ak, ' . ')
22  xtitle("Fourier coefficients of impulse train", "k", "
      ak")

```

Scilab code Exa 3.13 find fourier series coefficient of periodic signal

```

1  //Example 3_13
2  //Continuous Time Fourier Series Coefficients of a
   periodic signal  $x(t)=2+4\sin((5\pi)/3t)+\cos((2\pi/3)t)$ 
    $\pi/3)t$ 
3  clear;
4  clc;
5  t=0:0.01:1;
6  xt=2+4*sin((5*pi)/3*t)+cos((2*pi/3)*t);
7  x_t=2+4*sin((5*pi)/3*-t)+cos((2*pi/3)*-t);
8  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);
9  a0=1;
10 a2=(1/2)
11 a_2=(1/2)
12 a3=(4/(2*i));
13 a_3=-(4/(2*i));

```

```

14 ak=[zeros(1,5) a_3 a_2 0 a2 a3 zeros(1,5)];
15 k=-7:7;
16 disp("The fourier series coefficients are...")
17 disp(ak)
18 disp("magnitude of Fourier series coefficient")
19 disp(abs(ak))
20 subplot(2,1,1)
21 plot(k,abs(ak),'.');
22 xtitle("Magnitude Spectrum","k","jakj");
23 if xt==x_t then
24 disp("The Given signal is even. It has no phase
      spectrum");
25 else
26 phase=[zeros(1,6) ,%pi/2,0,-%pi/2,zeros(1,6)];
27 disp("Phase of Fourier series coefficient in radians
      ")
28 disp(phase)
29 subplot(2,1,2)
30 plot(k,phase,'.');
31 xtitle("Phase Spectrum","k","ak in radians");
32 end

```

Chapter 4

discrete time fourier series

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

```
1 //example 4_1
2 //determine the fourier series coefficient of the
   signal plot its magnitude and phase spectrum
3 clear;
4 close;
5 clc;
6 N = 10;
7 n = 0:0.01:N;
8 Wo = 2*%pi/N;
9 xn = ones(1, length(n)) + sin(Wo*n) + 3*cos(Wo*n) + cos(2*Wo
   *n+%pi/2);
10 for k = 0:N-2
11     C(k+1,:) = exp(-sqrt(-1)*Wo*n.*k);
12     a(k+1) = xn*C(k+1,:)' / length(n);
13     if(abs(a(k+1))) <= 0.1
14         a(k+1) = 0;
15     end
16 end
17 a = a';
18 a_conj = conj(a);
19 ak = [a_conj($:-1:1), a(2:$)];
```

```

20 Mag_ak = abs(ak);
21 for i = 1:length(a)
22     Phase_ak(i) = atan(imag(ak(i))/(real(ak(i))
        +0.0001));
23 end
24 Phase_ak = Phase_ak'
25 Phase_ak = [Phase_ak(1:$-1) -Phase_ak($:-1:1)];
26 k = -(N-2):(N-2);
27 //
28 figure
29 subplot(2,1,1)
30 a = gca();
31 a.y_location = "origin";
32 a.x_location = "origin";
33 plot2d3('ggn',k,real(ak),5)
34 poly1 = a.children(1).children(1);
35 poly1.thickness = 3;
36 title('Real part of(ak)')
37 xlabel('
        k')
38 subplot(2,1,2)
39 a = gca();
40 a.y_location = "origin";
41 a.x_location = "origin";
42 plot2d3('ggn',k,imag(ak),5)
43 poly1 = a.children(1).children(1);
44 poly1.thickness = 3;
45 title('imaginary part of(ak)')
46 xlabel('
        k')
47 //
48 figure
49 subplot(2,1,1)
50 a = gca();
51 a.y_location = "origin";
52 a.x_location = "origin";

```

```

53 plot2d3('gmn',k,Mag_ak,5)
54 poly1 = a.children(1).children(1);
55 poly1.thickness = 3;
56 title('abs(ak)')
57 xlabel('
    k')
58 subplot(2,1,2)
59 a = gca();
60 a.y_location = "origin";
61 a.x_location = "origin";
62 plot2d3('gmn',k,Phase_ak,5)
63 poly1 = a.children(1).children(1);
64 poly1.thickness = 3;
65 title('<(ak)')
66 xlabel('
    k')

```

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

```

1 //Example 4_2
2 //Finad DTFS coefficients of periodic square wave
3 clc;
4 clear;
5 N=10;
6 N1=2;
7 Wo=2*pi/N;
8 xn=ones(1,length(N));
9 n=-(2*N1+1):(2*N1+1);
10 a(1)=(2*N1+1)/N;
11 for k=1:2*N1
12 a(k+1)=sin((2*pi*k*(N1+0.5))/N)/sin(pi*k/N);
13 a(k+1)=a(k+1)/N;
14 if(abs(a(k+1))<=0.1)

```

```

15 a(k+1)=0;
16 end
17 end
18 a=a';
19 a_conj=conj(a);
20 ak=[a_conj($:-1:1),a(2:$)];
21 k=-2*N1:2*N1;
22 plot2d3('ggn',k,abs(ak))
23 xtitle('Magnitude spectrum','k','jakj')

```

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

```

1 //Example 4_3
2 //Discrete Time Fourier Transform coefficient of
   Periodic Impulse Train
3 clc;
4 clear;
5 N=5;
6 N1=-3*N:3*N;
7 xn=[zeros(1,N-1),1];
8 x=[1 xn xn xn xn xn xn ];
9 ak=1/N;
10 XW=2*pi*ak*ones(1,2*N);
11 Wo=2*pi/N;
12 n=-N:N-1;
13 W=Wo*n;
14 figure
15 subplot(2,1,1)
16 plot2d3('ggn',N1,x,2);
17 xtitle('Periodic Impulse Train','n','x[n]')
18 subplot(2,1,2)
19 plot2d3('ggn',W,XW,2);
20 xtitle('DTFT of Periodic Impulse Train','w','jX(exp(
   jw))j')
21 disp(Wo)

```

Scilab code Exa 4.12 determine DTFS

```
1 //Example 4_12_b
2 clc;
3 clear;
4 N=8;
5 n=0:0.01:N;
6 Wo=3*%pi/N;
7 xn=1*ones(1,length(n))+1*sin(Wo*n+%pi/4);
8 for k=0:N-2
9 C(k+1,:)=exp(-sqrt(-1)*Wo*n.*k);
10 a(k+1)=xn*C(k+1,:)/length(n);
11 if(abs(a(k+1))<=0.1)
12 a(k+1)=0;
13 end
14 end
15 a=a';
16 a_conj=conj(a);
17 ak=[a_conj($:-1:1),a(2:$)];
18 Mag_ak=abs(ak);
19 k=-(N-2):(N-2);
20 plot2d3('gnn',k,Mag_ak,5)
21 xtitle('abs(ak)','k','ak')
```

Scilab code Exa 4.14.a determine the fourier series coefficient

```
1 //Example 4_14_a
2 //determine the fourier series coefficient of x(n)
3 clc;
4 clear;
5 N=6;
```



```
6 n=0:0.01:N;
7 Wo=2*%pi/N;
8 xn=1*ones(1,length(n))+1*cos(Wo*n);
9 for k=0:N-2
10 C(k+1,:)=exp(-sqrt(-1)*Wo*n.*k);
11 a(k+1)=xn*C(k+1,:)/length(n);
12 if(abs(a(k+1))<=0.1)
13 a(k+1)=0;
14 end
15 end
16 a=a';
17 a_conj=conj(a);
18 ak=[a_conj($:-1:1),a(2:$)];
19 Mag_ak=abs(ak);
20 k=-(N-2):(N-2);
21 plot2d3('gnn',k,Mag_ak,5)
22 xtitle('abs(ak)', 'k', 'ak')
```

Chapter 5

continuous time fourier transform

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

```
1 //Example 5.1:
2 //Continuous Time Fourier Transform of a
3 clear;
4 clc;
5 close;
6 // Analog Signal
7 A =1; //Amplitude
8 Dt = 0.005;
9 t = 0:Dt:10;
10 xt=exp(-A*t);
11 //Continuous-time Fourier Transform
12 Wmax=2*%pi*1; //Analog Frequency = 1Hz
13 K=4;
14 k=0:(K/1000):K;
15 W=k*Wmax/K;
16 XW=xt*exp(-sqrt(-1)*t'*W)*Dt;
17 XW_Mag= abs(XW);
18 W=[-mtlbfliplr(W), W(2:1001)];
19 XW_Mag=[mtlbfliplr(XW_Mag),XW_Mag(2:1001)];
```

```

20 [XW_Phase ,db]=phasemag(XW);
21 XW_Phase=[-mtlbfliplr(XW_Phase),XW_Phase(2:1001)];
22 //Plotting Continuous Time Signal
23 a=gca();
24 a.y_location="origin";
25 plot(t,xt);
26 xlabel('t in sec. ');
27 ylabel('x(t)')
28 title('Continuous Time Signal')
29 //Plotting Magnitude Response of CTS
30 subplot(2,1,1);
31 a = gca();
32 a.y_location = "origin";
33 plot(W,XW_Mag);
34 xlabel('Frequency in Radians/Seconds—> W');
35 ylabel('abs(X(jW))')
36 title('Magnitude Response (CTFT)')
37 //Plotting Phase Reponse of CTS
38 subplot(2,1,2);
39 a = gca();
40 a.y_location = "origin";
41 a.x_location = "origin";
42 plot(W,XW_Phase*%pi/180);
43 xlabel('Frequency in Radians/Seconds—> W');
44 ylabel('<X(jW)')
45 title('Phase Response(CTFT) in Radians')

```

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

```

1 //Example 5.2:
2 //Continuous Time Fourier Transform of  $x(t)=e^{-a|t|}$ 
3 clc;
4 close;
5 a=1;
6 Dt=0.005;

```

```

7 t=-4.5:Dt:4.5;
8 xt=exp(-a*abs(t));
9 Wmax=2*%pi*1;
10 K=4;
11 k=0:(K/1000):K;
12 W=k*Wmax/K;
13 XW=xt*exp(-sqrt(-1)*t'*W)*Dt;
14 XW=real(XW);
15 W=[-mtlbfliplr(W), W(2:1001)];
16 XW=[mtlbfliplr(XW), XW(2:1001)];
17 subplot(1,1,1)
18 subplot(2,1,1);
19 a=gca();
20 a.y_location="origin";
21 plot(t,xt);
22 xlabel('t in sec. ');
23 ylabel('x(t) ');
24 title('Continuous Time Signal')
25 subplot(2,1,2);
26 a=gca();
27 a.y_location = "origin";
28 plot(W,XW);
29 xlabel('Frequency in Radians/Seconds W');
30 ylabel('X(jW) ');
31 title('Continuous-time Fourier Transform')

```

Scilab code Exa 5.6 determine fourier transform of signal

```

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

```

```

7 T0=4;
8 t=-T0/2:Dt:T0/2;
9 for i=1:length(t)
10 xt(i)=A;
11 end
12 Wmax=2*%pi*1;
13 K=4;
14 k=0:(K/1000):K;
15 W=k*Wmax/K;
16 xt=xt';
17 XW=xt*exp(-sqrt(-1)*t'*W)*Dt;
18 XW_Mag=real(XW);
19 W=[-mtlbfliplr(W),W(2:1001)];
20 XW_Mag=[mtlbfliplr(XW_Mag),XW_Mag(2:1001)];
21 subplot(2,1,1);
22 plot(t,xt);
23 xlabel('t in sec. ');
24 title('Contiuous Time Signal x(t)');
25 subplot(2,1,2);
26 plot(W,XW_Mag);
27 xlabel('Frequency in Radians/Seconds');
28 title('Continuous-time Fourier Transform X(jW)')

```

Scilab code Exa 5.8 find the fourier transform

```

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

```

```

11 k=0:(K/1000):K;
12 W=k*Wmax/K;
13 XW=xt*exp(-sqrt(-1)*t'*W)*Dt;
14 XW=real(XW);
15 W=[-mtlbfliplr(W),W(2:1001)];
16 XW=[mtlbfliplr(XW),XW(2:1001)];
17 subplot(2,1,1);
18 plot(t,xt);
19 xlabel('t i n s e c');
20 ylabel('x(t)');
21 title('Continuous Time Signal');
22 subplot(2,1,2);
23 plot(W,XW);
24 xlabel('Fr equency in Radians/Seconds W');
25 ylabel('X(jW)');
26 title('Continuous-time Fourier Transform')

```

Scilab code Exa 5.9 determine fourier transform of signal

```

1 //Example 5.9
2 //Continuous Time Fourier Transforms of cos(Wot)
3 clc;
4 close;
5 //CTFT
6 T1=2;
7 T=4*T1;
8 Wo=2*%pi/T;
9 W=[-Wo,0,Wo];
10 ak=(2*%pi*Wo*T1/%pi);
11 XW1=[-ak,0,ak];
12 figure
13 a = gca();
14 a.y_location="origin";
15 a.x_location="origin";
16 plot2d3('ggn',W,XW1,2);

```

```

17 poly1 = a.children(1).children(1);
18 poly1.thickness=3;
19 xlabel('W');
20 title('CTFT of cos(Wot)')

```

Scilab code Exa 5.10 determine fourier transform of signal

```

1 //Example 5.10
2 // Continuous Time Fourier Transforms of sin(Wot)
3 clc;
4 close;
5 //CTFT
6 T1=2;
7 T=4*T1;
8 Wo=2*pi/T;
9 W=[-Wo,0,Wo];
10 ak=(2*pi*Wo*T1/pi)/sqrt(-1);
11 XW=[-ak,0,ak];
12 //figure
13 a=gca();
14 a.y_location="origin";
15 a.x_location="origin";
16 plot2d3('ggn',W,imag(XW),2);
17 poly1=a.children(1).children(1);
18 poly1.thickness = 3;
19 xlabel('W');
20 title('CTFT of sin(Wot)')

```

Scilab code Exa 5.15 find the fourier transform

```

1 //Example 5_15
2 //Fourier Transform of  $x(t)=\exp(-t)*\sin(wc*t)*u(t)$ 
3 clear;

```

```

4  clc;
5  wc=1;
6  Dt=0.005;
7  t=0:Dt:10;
8  xt=(exp(t*(-1+wc))-exp(t*(-1-wc)))/(2*i);
9  Wmax=2*pi*1;
10 K=4;
11 k=0:(K/1000):K;
12 W=k*Wmax/K;
13 XW=xt*exp(-sqrt(-1)*t'*W)*Dt;
14 XW_Mag=abs(XW);
15 W=[-mtlbfliplr(W),W(2:1001)];
16 XW_Mag=[mtlbfliplr(XW_Mag),XW_Mag(2:1001)];
17 [XW_Phase,db]=phasemag(XW);
18 XW_Phase=[-mtlbfliplr(XW_Phase),XW_Phase(2:1001)];
19 //Plotting Continuous Time Signal
20 figure(1)
21 plot(t,xt);
22 xlabel('t in sec. ');
23 ylabel('x(t)');
24 title('Continuous Time Signal')
25 figure(2)
26 //Plotting Magnitude Response of CTS
27 subplot(2,1,1);
28 plot(W,XW_Mag);
29 xlabel('Frequency in Radians/Seconds>W');
30 ylabel('abs(X(jW))');
31 title('Magnitude Response (CTFT)');
32 //Plotting Phase Reponse of CTS
33 subplot(2,1,2);
34 plot(W,XW_Phase*pi/180);
35 xlabel('Frequency in Radians/Seconds>W');
36 ylabel('<X(jW)');
37 title('Phase Response (CTFT) in Radians')

```

Scilab code Exa 5.16 find the fourier transform

```
1 //Example 5_16
2 //Fourier Transform of  $x(t)=\exp(-a*t)*\cos(\omega*c*t)*u(t)$ 
3 clear;
4 clc;
5 a=1;
6  $\omega c=1$ ;
7 Dt=0.005;
8 t=0: Dt :10;
9  $xt=(\exp(t*(-a+\omega c))+\exp(t*(-a-\omega c)))/2$ ;
10  $W_{max}=2*\%pi*1$ ;
11 K=4;
12 k=0:(K/1000):K;
13  $W=k*W_{max}/K$ ;
14  $XW=xt*\exp(-\sqrt{-1}*t'*W)*Dt$ ;
15  $XW\_Mag=abs(XW)$ ;
16  $W=[-mtlb\_fliplr(W),W(2:1001)]$ ;
17  $XW\_Mag=[mtlb\_fliplr(XW\_Mag),XW\_Mag(2:1001)]$ ;
18  $[XW\_Phase,db]=phasemag(XW)$ ;
19  $XW\_Phase=[-mtlb\_fliplr(XW\_Phase),XW\_Phase(2:1001)]$ ;
20 //Plotting Continuous Time Signal
21 figure(1)
22 plot(t,xt);
23 xlabel('t in sec. ');
24 ylabel('x(t)')
25 title('Continuous Time Signal')
26 figure(2)
27 //Plotting Magnitude Response of CTS
28 subplot(2,1,1);
29 plot(W,XW_Mag);
30 xlabel('Frequency in Radians/Seconds>W');
31 ylabel('abs(X(jW))')
32 title('Magnitude Response (CTFT)')
33 //Plotting Phase Reponse of CTS
34 subplot(2,1,2);
35 plot(W,XW_Phase*%pi/180);
36 xlabel('Frequency in Radians/Seconds>W');
```

```

37 ylabel('∠X(jW)')
38 title('Phase Response (CTFT) in Radians')

```

Scilab code Exa 5.17 find the fourier transform

```

1 //Example 5_17
2 //Fourier Transform of Continuous Time Signal x(t)=
   cos(wc*t)*u(t)
3 clear;
4 clc;
5 wc=1;
6 Dt=0.005;
7 t=0: Dt :10;
8 xt=(exp(wc*t)+exp(-wc*t))/2;
9 Wmax=2*%pi*1;
10 K=4;
11 k=0:(K/1000):K;
12 W=k*Wmax/K;
13 XW=xt*exp(-sqrt(-1)*t'*W)*Dt;
14 XW_Mag=abs(XW);
15 W=[-mtlb_fliplr(W),W(2:1001)];
16 XW_Mag=[mtlb_fliplr(XW_Mag),XW_Mag(2:1001)];
17 [XW_Phase,db]=phasemag(XW);
18 XW_Phase=[-mtlb_fliplr(XW_Phase),XW_Phase(2:1001)];
19 //Plotting Continuous Time Signal
20 figure(1)
21 plot(t,xt);
22 xlabel('t in sec. ');
23 ylabel('x(t)')
24 title('Continuous Time Signal')
25 figure(2)
26 //Plotting Magnitude Response of CTS
27 subplot(2,1,1);
28 plot(W,XW_Mag);
29 xlabel('Frequency in Radians/Seconds>W');

```

```

30 ylabel('abs(X(jW))')
31 title('Magnitude Response (CTFT)')
32 //Plotting Phase Reponse of CTS
33 subplot(2,1,2);
34 plot(W,XW_Phase*%pi/180);
35 xlabel('Frequency in Radians/Seconds>W');
36 ylabel('<X(jW)')
37 title('Phase Response (CTFT) in Radians')

```

Scilab code Exa 5.26 find and sketch fourier transform

```

1 //Example 5_26
2 //find and sketch Fourier Transform of Periodic
   Impulse Train
3 clear;
4 clc;
5 T=-4:4;;
6 T1=1;
7 xt=ones(1,length(T));
8 ak=1/T1;
9 XW=2*%pi*ak*ones(1,length(T));
10 Wo=2*%pi/T1;
11 W=Wo*T;
12 figure
13 subplot(2,1,1)
14 plot2d3('gnn',T,xt);
15 xlabel('t');
16 title('Periodic Impulse Train')
17 subplot(2,1,2)
18 plot2d3('gnn',W,XW);
19 xlabel('t');
20 title('CTFT of Periodic Impulse Train')

```

Chapter 6

discrete time fourier transform

Scilab code Exa 6.2 find fourier transform of causal sequence

```
1 //Example 6.2:
2 //Discrete Time Fourier Transform of discrete
   sequence  $x[n] = (a^n) \cdot u[n]$ ,  $a > 0$  and  $a < 0$ 
3 clear;
4 clc;
5 close;
6 // DTS Signal
7 a1 = 0.5;
8 a2 = -0.5;
9 max_limit = 10;
10 for n = 0:max_limit-1
11     x1(n+1) = (a1^n);
12     x2(n+1) = (a2^n);
13 end
14 n = 0:max_limit-1;
15 Wmax = 2*%pi;
16 K = 4;
17 k = 0:(K/1000):K;
18 W = k*Wmax/K;
19 x1 = x1';
20 x2 = x2';
```

```

21 XW1 = x1* exp(-sqrt(-1)*n'*W);
22 XW2 = x2* exp(-sqrt(-1)*n'*W);
23 XW1_Mag = abs(XW1);
24 XW2_Mag = abs(XW2);
25 W = [-mtlbfliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
26 XW1_Mag = [mtlbfliplr(XW1_Mag), XW1_Mag(2:1001)];
27 XW2_Mag = [mtlbfliplr(XW2_Mag), XW2_Mag(2:1001)];
28 [XW1_Phase,db] = phasemag(XW1);
29 [XW2_Phase,db] = phasemag(XW2);
30 XW1_Phase = [-mtlbfliplr(XW1_Phase),XW1_Phase
    (2:1001)];
31 XW2_Phase = [-mtlbfliplr(XW2_Phase),XW2_Phase
    (2:1001)];
32 figure
33 subplot(3,1,1);
34 plot2d3('gmn',n,x1);
35 xtitle('Discrete Time Sequence x[n] for a>0')
36 subplot(3,1,2);
37 a = gca();
38 a.y_location = "origin";
39 a.x_location = "origin";
40 plot2d(W,XW1_Mag);
41 title('Magnitude Response abs(X(jW))')
42 subplot(3,1,3);
43 a = gca();
44 a.y_location = "origin";
45 a.x_location = "origin";
46 plot2d(W,XW1_Phase);
47 title('Phase Response <(X(jW))')
48 //plot for a<0
49 figure
50 subplot(3,1,1);
51 plot2d3('gmn',n,x2);
52 xtitle('Discrete Time Sequence x[n] for a>0')
53 subplot(3,1,2);

```

Scilab code Exa 6.5 find fourier transform of noncausal sequence

```
1 //Example 6.5:
2 //Find The Fourier Transform
3 clc;
4 a = 0.5;
5 max_limit = 10;
6 n = -max_limit+1:max_limit-1;
7 x = a^abs(n);
8 Wmax = 2*%pi;
9 K = 4;
10 k = 0:(K/1000):K;
11 W = k*Wmax/K;
12 XW = x* exp(-sqrt(-1)*n'*W);
13 XW_Mag = real(XW);
14 W = [-mtlbfliplr(W), W(2:1001)];
15 XW_Mag = [mtlbfliplr(XW_Mag), XW_Mag(2:1001)];
16 //plot for abs(a)<1
17 figure
18 subplot(2,1,1);
19 a = gca();
20 a.y_location = "origin";
21 a.x_location = "origin";
22 plot2d3('gnn',n,x);xtitle('Discrete Time Sequence x[
    n] for a>0')
23 subplot(2,1,2);
24 a = gca();
25 a.y_location = "origin";
26 a.x_location = "origin";
27 plot2d(W,XW_Mag);
28 title('Discrete Time Fourier Transform X(exp(jW))')
```

Scilab code Exa 6.6 find fourier transform of rectangular pulse

```
1 //Example 6.6:
2 //Discrete Time Fourier Transform of
3 clc;
4 // DTS Signal
5 N1 = 2;
6 n = -N1:N1;
7 x = ones(1,length(n));
8 // Discrete-time Fourier Transform
9 Wmax = 2*%pi;
10 K = 4;
11 k = 0:(K/1000):K;
12 W = k*Wmax/K;
13 XW = x* exp(-sqrt(-1)*n'*W);
14 XW_Mag = real(XW);
15 W = [-mtlbfliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
16 XW_Mag = [mtlbfliplr(XW_Mag), XW_Mag(2:1001)];
17 //plot for abs(a)<1
18 figure
19 subplot(2,1,1);
20 a = gca();
21 a.y_location = "origin";
22 a.x_location = "origin";
23 plot2d3('gnn',n,x);
24 xtitle('Discrete Time Sequence x[n]')
25 subplot(2,1,2);
26 a = gca();
27 a.y_location = "origin";
28 a.x_location = "origin";
29 plot2d(W,XW_Mag);
30 title('Discrete Time Fourier Transform X(exp(jW))')
```

Scilab code Exa 6.12 find fourier transform of exponential sequence

```

1 //Example 6_12:
2 //Discrete Time Fourier Transform of discrete
   sequence  $x[n]=((-1)^n)*(a^n).u[n]$ ,  $a>0$  and  $a<0$ 
3 clear;
4 clc;
5 close;
6 //DTS Signal
7 a1=0.5;
8 a2=-0.5;
9 max_limit=10;
10 for n=0:max_limit-1
11 x1(n+1)=((-1)^n)*(a1^n);
12 x2(n+1)=((-1)^n)*(a2^n);
13 end
14 n=0:max_limit-1;
15 Wmax=2*%pi;
16 K=4;
17 k=0:(K/1000):K;
18 W=k*Wmax/K;
19 x1=x1';
20 x2=x2';
21 XW1=x1*exp(-sqrt(-1)*n'*W);
22 XW2=x2*exp(-sqrt(-1)*n'*W);
23 XW1_Mag=abs(XW1);
24 XW2_Mag=abs(XW2);
25 W=[-mtlbfliplr(W),W(2:1001)];
26 XW1_Mag = [mtlbfliplr(XW1_Mag),XW1_Mag(2:1001)];
27 XW2_Mag = [mtlbfliplr(XW2_Mag), XW2_Mag(2:1001)];
28 [XW1_Phase,db] = phasemag(XW1);
29 [XW2_Phase,db] = phasemag(XW2);
30 XW1_Phase = [-mtlbfliplr(XW1_Phase),XW1_Phase
   (2:1001)];
31 XW2_Phase = [-mtlbfliplr(XW2_Phase),XW2_Phase
   (2:1001)];
32 figure
33 subplot(3,1,1);
34 plot2d3('gmn',n,x1);
35 xtitle('Discrete Time Sequence  $x[n]$  for  $a>0$ ')

```



```

36 subplot(3,1,2);
37 a = gca();
38 a.y_location = "origin";
39 a.x_location = "origin";
40 plot2d(W,XW1_Mag);
41 title('Magnitude Response abs(X(jW))')
42 subplot(3,1,3);
43 a = gca();
44 a.y_location = "origin";
45 a.x_location = "origin";
46 plot2d(W,XW1_Phase);
47 title('Phase Response <(X(jW))')
48 //plot for a<0
49 figure
50 subplot(3,1,1);
51 plot2d3('gmn',n,x2);
52 xtitle('Discrete Time Sequence x[n] for a>0')
53 subplot(3,1,2);

```

Scilab code Exa 6.13 determine fourier transform of sequence

```

1 //Example 6_13:
2 //Discrete Time Fourier Transform of discrete
   sequence  $x[n]=(n)*(a^n).u[n]$ ,  $a>0$  and  $a<0$ 
3 clear;
4 clc;
5 close;
6 //DTS Signal
7 a1=0.5;
8 a2=-0.5;
9 max_limit=10;
10 for n=0:max_limit-1
11 x1(n+1)=(n)*(a1^n);
12 x2(n+1)=(n)*(a2^n);
13 end

```

```

14 n=0:max_limit-1;
15 Wmax=2*%pi;
16 K=4;
17 k=0:(K/1000):K;
18 W=k*Wmax/K;
19 x1=x1';
20 x2=x2';
21 XW1=x1*exp(-sqrt(-1)*n'*W);
22 XW2=x2*exp(-sqrt(-1)*n'*W);
23 XW1_Mag=abs(XW1);
24 XW2_Mag=abs(XW2);
25 W=[-mtlbfliplr(W),W(2:1001)];
26 XW1_Mag = [mtlbfliplr(XW1_Mag),XW1_Mag(2:1001)];
27 XW2_Mag = [mtlbfliplr(XW2_Mag), XW2_Mag(2:1001)];
28 [XW1_Phase,db] = phasemag(XW1);
29 [XW2_Phase,db] = phasemag(XW2);
30 XW1_Phase = [-mtlbfliplr(XW1_Phase),XW1_Phase
               (2:1001)];
31 XW2_Phase = [-mtlbfliplr(XW2_Phase),XW2_Phase
               (2:1001)];
32 figure
33 subplot(3,1,1);
34 plot2d3('gmn',n,x1);
35 xtitle('Discrete Time Sequence x[n] for a>0')
36 subplot(3,1,2);
37 a = gca();
38 a.y_location = "origin";
39 a.x_location = "origin";
40 plot2d(W,XW1_Mag);
41 title('Magnitude Response abs(X(jW))')
42 subplot(3,1,3);
43 a = gca();
44 a.y_location = "origin";
45 a.x_location = "origin";
46 plot2d(W,XW1_Phase);
47 title('Phase Response <(X(jW))')
48 //plot for a<0
49 figure

```

```

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

```

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

```

1 //Example6.18:
2 //Find and sketch the Fourier Transform
3 clc;
4 N = 5;
5 N1 = -3*N:3*N;
6 xn = [zeros(1,N-1),1];
7 x = [1 xn xn xn xn xn xn];
8 ak = 1/N;
9 XW = 2*%pi*ak*ones(1,2*N);
10 Wo = 2*%pi/N;
11 n = -N:N-1;
12 W = Wo*n;
13 figure
14 subplot(2,1,1)
15 a = gca();
16 a.y_location = "origin";
17 a.x_location = "origin";
18 plot2d3('gnn',N1,x,2);
19 poly1 = a.children(1).children(1);
20 poly1.thickness = 3;
21 xlabel('n');
22 title('Periodic Impulse Train')
23 subplot(2,1,2)
24 a = gca();
25 a.y_location = "origin";
26 a.x_location = "origin";
27 plot2d3('gnn',W,XW,2);
28 poly1 = a.children(1).children(1);

```

```

29 poly1.thickness = 3;
30 xlabel('W');
31 title('DTFT of Periodic Impulse Train')
32 disp(Wo)

```

Scilab code Exa 6.19 find fourier transform of periodic signal

```

1 //Example6.19:
2 //Discrete Time Fourier Transform
3 clc;
4 N = 5;
5 Wo = 2*%pi/N;
6 W = [-Wo,0,Wo];
7 XW =[%pi,0,%pi]; //figure
8 a = gca();
9 a.y_location = "origin";
10 a.x_location = "origin";
11 plot2d3('gnn',W,XW,2);
12 poly1 = a.children(1).children(1);
13 poly1.thickness = 3;
14 xlabel('W');
15 title('DTFT of cos(nWo)')
16 disp(Wo)

```

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

```

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

```

```

7 n = 0:0.01:N;
8 Wo = 2*%pi/N;
9 xn =sin(Wo*n);
10 for k =0:N-2
11     C(k+1,:) = exp(sqrt(-1)*Wo*n.*k);
12     a(k+1) = xn*C(k+1,:)/length(n);
13     if(abs(a(k+1))<=0.1)
14         a(k+1)=0;
15     end
16 end
17 a =a';
18 a_conj =conj(a);
19 ak = [a_conj($:-1:1),a(2:$)];
20 Mag_ak = abs(ak);
21 k = -(N-2):(N-2);
22 plot2d3('gnn',k,Mag_ak,5)
23 xtitle('abs(ak)', 'k', 'ak')

```

Chapter 8

sampling

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

```
1 //determine the nyquist rate
2 //8.1(a)
3 clc;
4 clear all;
5 //x(t)=sin(200*pi*t)
6 wp=200;
7 F1=wp/2;
8 Fs=2*F1;
9 disp('Nyquist Rate=');
10 disp(Fs);
```

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

```
1 //determine the nyquist rate
2 //8.1(b)
3 clc;
4 clear all;
5 //x(t)=sin2(200*pi*t)
```

```

6 //x(t)=0.5-0.5cos(400*pi*t)
7 wp=400;
8 F1=wp/2;
9 Fs=2*F1;
10 disp('Nyquist Rate=');
11 disp(Fs);

```

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

```

1 //determine the nyquist rate
2 //example 8_1<c>
3 clc;
4 clear all;
5 //x(t)=1+cos(200*pi*t)+sin(400*pi*t)
6 wq=200;
7 wp=400;
8 wf=0;
9 if wp>=wq then
10 wf=wp;
11 else
12 wf=wq;
13 end
14 F1=wf/2;
15 Fs=2*F1;
16 disp('Nyquist Rate=');
17 disp(Fs);

```

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

```

1 //determine the nyquist rate
2 //example 8_1<d>
3 clc;
4 clear all;

```

```

5 //x(t)=cos(150*pi*t)sin(100*pi*t)
6 //x(t)=0.5sin(250*pi*t)*0.5*sin(50*pi*t)
7 wq=50;
8 wp=250;
9 wf=0;
10 if wp>=wq then
11 wf=wp;
12 else
13 wf=wq;
14 end
15 F1=wf/2;
16 Fs=2*F1;
17 disp('Nyquist Rate=');
18 disp(Fs);

```

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

```

1 //determine the nyquist rate
2 //example 8_1<e>
3 clc;
4 clear all;
5 //x(t)=cos3(200*pi*t)
6 //cos3(t)=1/4[3cos(t)+cos(3t)]
7 //cos3(t)=3/4[cos(200)+1/4cos(600)]
8 wq=600;
9 wp=200;
10 wf=0;
11 if wp>=wq then
12 wf=wp;
13 else
14 wf=wq;
15 end
16 F1=wf/2;
17 Fs=2*F1;
18 disp('Nyquist Rate=');

```



```
19 disp(Fs);
```

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

```
1 //Example 8_2 <a>
2 //determine the nyquist rate of  $x(t)=\text{sinc}(200*\pi*t)$ 
3 //  $\text{sinc}(t)=\cos(t)/t$ 
4 //  $\cos^3(t)=3/4[\cos(200)+1/4\cos(600)]$ 
5 clc;
6 clear all;
7 wp=200;
8 F1=wp/2;
9 Fs=2*F1;
10 disp('Nyquist Rate=');
11 disp(Fs);
```

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

```
1 //Example 8_2<b>
2 //determine the nyquist rate of  $x(t)=\text{sinc}^2(200*\pi*t)$ 
3 //  $\text{sinc}(400t)=0.5\cos(400t)/400t$ 
4 clc;
5 clear all;
6 wp=400;
7 F1=wp/2;
8 Fs=2*F1;
9 disp('Nyquist Rate=');
10 disp(Fs);
```

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

```

1 //Example 8_2<c>
2 //determine the nyquist rate of  $x(t)=\text{sinc}(200\pi t)+\text{sinc}^2(200\pi t)$ 
3 //here ,  $\text{sinc}(400t)=0.5\cos(400t)/400t+$ 
4 clc;
5 clear all;
6 wq=400;
7 wp=200;
8 wf=0;
9 if wp>=wq then
10 wf=wp;
11 else
12 wf=wq;
13 end
14 F1=wf/2;
15 Fs=2*F1;
16 disp('Nyquist Rate=');
17 disp(Fs);

```

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

```

1 //Example 8_3 <a>
2 //determine whether the Nyquist criteria satisfy or
  not
3 //Ws>=2Wmax
4 //fs>=2fmax
5 clc;
6 clear all;
7 Ts=0.5*10-3;
8 Wc=1000
9 Fs=1000
10 Ts_test=1/Fs;
11 if (Ts<=Ts_test) then
12 disp('Nyquist Criteria Satisfy')
13 else

```

```
14 disp('Nyquist Criteria NOT Satisfy ');
15 end
```

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

```
1 //Example 8_3 <b>
2 //determine whether the Nyquist criteria satisfy or
  not
3 //Ws>=2Wmax
4 //fs>=2fmax
5 clc;
6 clear all;
7 Ts=2*10^-3;
8 Wc=1000
9 Fs=1000
10 Ts_test=1/Fs;
11 if (Ts<=Ts_test) then
12 disp('Nyquist Criteria Satisfy ')
13 else
14 disp('Nyquist Criteria NOT Satisfy ');
15 end
```

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

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

```

9 Fs=1000
10 Ts_test=1/Fs;
11 if (Ts<=Ts_test) then
12 disp('Nyquist Criteria Satisfy')
13 else
14 disp('Nyquist Criteria NOT Satisfy ');
15 end

```

Scilab code Exa 8.9.a determine minimum sampling frequency

```

1 //Example 8_9 <a>
2 //Determine minimum sampling frequency
3 clc;
4 clear all;
5 Fl=9000;
6 Fh=12000;
7 Bandwidth_1=Fh-Fl;
8 a=modulo(Fh,Bandwidth_1);
9 Fh_1=Fh-a;
10 div_12=Fh_1./Bandwidth_1;
11 if(a==0) then
12 Fs=2*Bandwidth_1;
13 else
14 Fs=(2*Fh)/div_12;
15 end
16 disp('Minimum Sampling Frequency=');
17 disp(Fs);

```

Scilab code Exa 8.9.b determine minimum sampling frequency

```

1 //Example 8_9 <b>
2 //Determine minimum sampling frequency
3 clc;

```

```

4 clear all;
5 Fl=18000;
6 Fh=22000;
7 Bandwidth_1=Fh-Fl;
8 a=modulo(Fh,Bandwidth_1);
9 Fh_1=Fh-a;
10 div_12=Fh_1./Bandwidth_1;
11 if(a==0) then
12 Fs=2*Bandwidth_1;
13 else
14 Fs=(2*Fh)/div_12;
15 end
16 disp('Minimum Sampling Frequency=');
17 disp(Fs);

```

Scilab code Exa 8.9.c determine minimum sampling frequency

```

1 //Example 8_9 <c>
2 //Determine minimum sampling frequency
3 clc;
4 clear all;
5 Fl=30000;
6 Fh=35000;
7 Bandwidth_1=Fh-Fl;
8 a=modulo(Fh,Bandwidth_1);
9 Fh_1=Fh-a;
10 div_12=Fh_1./Bandwidth_1;
11 if(a==0) then
12 Fs=Bandwidth_1;
13 else
14 Fs=Bandwidth_1;
15 end
16 disp('Minimum Sampling Frequency=');
17 disp(Fs);

```

Scilab code Exa 8.10.a determine minimum sampling rate

```
1 //Example 8_10 <a>
2 //determine the Nyquist rate
3 //x(t)=10cos(2000)cos(8000)
4 //x(t)=5cos(6000)+5cos(10000)
5 clc;
6 clear all;
7 wq=10000;
8 wp=6000;
9 wf=0;
10 if wp>=wq then
11 wf=wp;
12 else
13 wf=wq;
14 end
15 F1=wf/2;
16 Fs=2*F1;
17 disp('Nyquist Rate=');
18 disp(Fs);
```

Scilab code Exa 8.10.b determine minimum sampling rate

```
1 //Example 8_10 <b>
2 //determine the minimum sampling rate
3 //x(t)=10cos(2000)cos(8000)
4 //x(t)=5cos(6000)+5cos(10000)
5 clc;
6 clear all;
7 F1=6000/2;
8 Fh=10000/2;
9 Bandwidth_1=Fh-F1;
```

```
10 a=modulo(Fh,Bandwidth_1);
11 Fh_1=Fh-a;
12 div_12=Fh_1./Bandwidth_1;
13 if(a==0) then
14 Fs=2*Bandwidth_1;
15 else
16 Fs=(2*Fh)/div_12;
17 end
18 disp('Minimum Sampling Frequency=');
19 disp(Fs);
```

Chapter 10

z transform

Scilab code Exa 10.12.a find z transform

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

Scilab code Exa 10.12.b find z transform

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



```

4 function [za]=ztransfer(sequence,n)
5 z=poly(0,'z','r')
6 za=sequence*(1/z)^n'
7 endfunction
8 x=[1,2,6,-2,0,3];
9 n1=-2:length(x)-3;
10 X=ztransfer(x,n1);
11 disp(X,'X(z)=');
12 funcprot(0);

```

Scilab code Exa 10.12.c find z transform

```

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

```

Scilab code Exa 10.12.d find z transform

```

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

```

```

7  endfunction
8  x=[1,2,6,-2,0,3];
9  n1=-5:length(x)-6;
10 X=ztransfer(x,n1);
11 disp(X,'X(z)=');
12 funcprot(0);

```

Scilab code Exa 10.27 find the convolution of the given signal

```

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

```

Scilab code Exa 10.33 find inverse z transform

```

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

```

Scilab code Exa 10.34 determine inverse z transform

```

1 //Example 10_34

```

```

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

```

Scilab code Exa 10.41 find the inverse z transform

```

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

```

Scilab code Exa 10.53.a determine the unilateral z transform

```

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

```

Scilab code Exa 10.53.b determine unilateral z transform

```
1 //example 10_53<b>:
2 //Find unilateral Z transform
3 clc;
4 function [za]=ztransfer(sequence,n)
5 z=poly(0,'z','r')
6 za=sequence*(1/z)^n'
7 endfunction
8 x=[1,2,5,4,0,3];
9 n1=-2:3;
10 count=0;
11 for a=n1(1):length(x)
12     if a==0 then
13         abc=count;
14     else
15     end
16     count=count+1;
17 end
18 abc=abc+1;
19 ac1=0;
20 x11=[1 1 1 1];
21 for a=abc:length(x)
22     ac1=ac1+1;
23     x11(ac1)=x(a);
24 end
25 n11=0:(length(x)-abc);
26 X=ztransfer(x11,n11);
27 disp(X,'X(z)=');
28 funcprot(0);
```

Scilab code Exa 10.53.c determine unilateral z transform

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

Chapter 11

state space analysis

Scilab code Exa 11.9 find state space representation of the system

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