

Scilab Textbook Companion for
Signals And Systems
by P. R. Rao¹

Created by
Satyam Agrawal
B.Tech
Instrumentation Engineering
Krishna Institute Of Engineering and Technology
College Teacher
Varun Gupta
Cross-Checked by
Chaitanya

June 24, 2014

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: P. R. Rao

Publisher: TMH Education Pvt. Ltd., New Delhi

Edition: 5

Year: 2011

ISBN: 978-0-07-066927-7

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 Signals Continuous and Discrete time	12
2 Laplace and Z Transform	34
3 Fourier Series of Continuous Time Signals	49
4 The Continuous Time Fourier Transform	62
5 Fourier Representation of Discrete Time Signals	85
6 Sampling and Reconstruction of Bandlimited Signals	107
7 Systems	119
8 Systems Time and Frequency Domain Analysis	149
9 Convolution and Correlation Continuous Time Signals	164
10 Discrete Time Convolution and Correlation	180
11 Hilbert Transform Continuous and Discrete	190

List of Scilab Codes

Exa 1.1	Finding Fundamental Period	12
Exa 1.3.a	Periodicity	13
Exa 1.3.b	Periodicity and fundamental period	15
Exa 1.4	Fundamental Period	17
Exa 1.5	Fundamental Period	18
Exa 1.6	Fundamental Period	20
Exa 1.7	Even and odd Components	21
Exa 1.11	Waveforms	23
Exa 1.12	Time Scaling	29
Exa 1.13	Plot at a particular sampling frequency	29
Exa 2.1	Laplace Transform	34
Exa 2.2	Inverse Laplace	34
Exa 2.3	Initial Value Theorem	35
Exa 2.4	Laplace Transform	36
Exa 2.6	Circuit Current	36
Exa 2.7	Inverse Laplace Transform	36
Exa 2.8	Inverse Laplace Transform	37
Exa 2.9	Circuit Current	38
Exa 2.10	Laplace Transform	38
Exa 2.15	Laplace Transform	39
Exa 2.16	Laplace Transform	40
Exa 2.17	Laplace Transform	40
Exa 2.19	Laplace Transform	42
Exa 2.21	Unilateral Z Transform	43
Exa 2.29	Z Transform	44
Exa 2.37	Inverse Z Transform	44
Exa 2.38.a	Inverse Z Transform	45
Exa 2.38.b	Inverse Z Transform	46

Exa 2.39	Inverse Z Transform	47
Exa 2.40	Inverse Z Transform	47
Exa 3.13	Complex Fourier Exponential Series	49
Exa 3.14	Complex Fourier Exponential Series	55
Exa 3.16	Complex Fourier Exponential Series	56
Exa 3.17	Complex Fourier Exponential Series	60
Exa 3.18	Trigonometric Series	61
Exa 4.1	Magnitude and Phase Spectra	62
Exa 4.2	Magnitude and Phase Spectra	66
Exa 4.3	Magnitude and Phase Spectra	71
Exa 4.4	Energy at output	72
Exa 4.5	Fourier Transform	72
Exa 4.6	Convolution	75
Exa 4.7	Energy in Signal	77
Exa 4.11	Fourier Transform	78
Exa 4.14	Fourier Transform	78
Exa 4.17	Fourier Transform	81
Exa 5.1	DTFT	85
Exa 5.2	DTFT and Spectra	86
Exa 5.3	Inverse Fourier	89
Exa 5.4	Inverse Fourier	92
Exa 5.5	Fourier series	95
Exa 5.9.i	DTFT	98
Exa 5.9.ii	DTFT	99
Exa 5.9.iii	DTFT	100
Exa 5.11	Inverse Fourier	101
Exa 5.13	Circular Convolution	102
Exa 5.14	Circular Convolution	104
Exa 5.16	DTFT	105
Exa 5.18	DFT coefficients	106
Exa 6.1	Sampling and Reconstruction	107
Exa 6.2	Sampling and Reconstruction	108
Exa 6.3	Minimum Number of samples	112
Exa 6.4	Minimum Sampling Frequency	114
Exa 6.5	LPF output	114
Exa 6.6	Sampling Frequency	116
Exa 6.8	Minimum Sampling Frequency	118
Exa 7.1	Properties of System	119

Exa 7.3	System Properties	120
Exa 7.4	System Properties	121
Exa 7.5	System Properties	122
Exa 7.6	System Properties	123
Exa 7.7	Properties of System	124
Exa 7.8	Properties of System	125
Exa 7.9	Properties of System	126
Exa 7.10	Output of an LTI	127
Exa 7.11	Convolution	127
Exa 7.14	Impulse and Step response	129
Exa 7.15	Impulse and Step Response	131
Exa 7.19	Convolution	134
Exa 7.22	Convolution	137
Exa 7.23	Convolution	140
Exa 7.24	Step Response	141
Exa 7.25	Step response	143
Exa 7.33	Magnitude and Phase Response	146
Exa 8.1	Magnitude and Phase Responses	149
Exa 8.2	Transfer Function of system	152
Exa 8.3	Magnitude and Phase Responses	152
Exa 8.4	system response	155
Exa 8.5	System function	157
Exa 8.8	Transfer function	159
Exa 8.9	Transfer function by block diagram reduction	161
Exa 8.10	Realization Structure	161
Exa 8.12	Transfer function Realization	162
Exa 8.13	Parallel form realization	163
Exa 9.1	Convolution	164
Exa 9.3	Convolution	166
Exa 9.4	Autocorrelation	167
Exa 9.5	Autocorrelation	169
Exa 9.6	Autocorrelation	171
Exa 9.7	Autocorrelation	174
Exa 9.10	Autocorrelation Function	177
Exa 9.11	PSD	178
Exa 10.1	Cross Correlation	180
Exa 10.2	Cross Correlation Sequence	182
Exa 10.4	Auto Correlation	184

Exa 10.5	Auto Correlation	184
Exa 10.8	Auto Correlation	186
Exa 10.11	System response	187
Exa 11.1	Hilbert Transform	190
Exa 11.2	Hilbert Transform	193
Exa 11.3	Hilbert Transform	195
Exa 11.6	Hilbert Transform	198

List of Figures

1.1	Finding Fundamental Period	13
1.2	Finding Fundamental Period	14
1.3	Periodicity	16
1.4	Periodicity and fundamental period	17
1.5	Fundamental Period	19
1.6	Fundamental Period	20
1.7	Fundamental Period	22
1.8	Even and odd Components	24
1.9	Even and odd Components	25
1.10	Waveforms	27
1.11	Waveforms	28
1.12	Time Scaling	30
1.13	Time Scaling	31
1.14	Plot at a particular sampling frequency	33
2.1	Laplace Transform	39
2.2	Laplace Transform	41
2.3	Laplace Transform	42
2.4	Laplace Transform	43
3.1	Complex Fourier Exponential Series	50
3.2	Complex Fourier Exponential Series	51
3.3	Complex Fourier Exponential Series	53
3.4	Complex Fourier Exponential Series	54
3.5	Complex Fourier Exponential Series	56
3.6	Complex Fourier Exponential Series	58
3.7	Complex Fourier Exponential Series	59
4.1	Magnitude and Phase Spectra	63

4.2	Magnitude and Phase Spectra	64
4.3	Magnitude and Phase Spectra	66
4.4	Magnitude and Phase Spectra	67
4.5	Magnitude and Phase Spectra	69
4.6	Magnitude and Phase Spectra	70
4.7	Fourier Transform	73
4.8	Fourier Transform	74
4.9	Convolution	76
4.10	Fourier Transform	79
4.11	Fourier Transform	80
4.12	Fourier Transform	82
4.13	Fourier Transform	83
5.1	DTFT	87
5.2	DTFT	88
5.3	DTFT and Spectra	90
5.4	DTFT and Spectra	91
5.5	Inverse Fourier	93
5.6	Inverse Fourier	94
5.7	Inverse Fourier	96
5.8	Inverse Fourier	97
5.9	Inverse Fourier	102
5.10	Inverse Fourier	103
6.1	Sampling and Reconstruction	109
6.2	Sampling and Reconstruction	110
6.3	Sampling and Reconstruction	112
6.4	Sampling and Reconstruction	113
6.5	LPF output	116
6.6	LPF output	117
7.1	Output of an LTI	128
7.2	Convolution	130
7.3	Impulse and Step response	132
7.4	Impulse and Step response	133
7.5	Impulse and Step Response	135
7.6	Impulse and Step Response	136
7.7	Convolution	138

7.8	Convolution	140
7.9	Convolution	142
7.10	Step Response	144
7.11	Step response	145
7.12	Magnitude and Phase Response	147
7.13	Magnitude and Phase Response	148
8.1	Magnitude and Phase Responses	150
8.2	Magnitude and Phase Responses	151
8.3	Transfer Function of system	153
8.4	Magnitude and Phase Responses	155
8.5	Magnitude and Phase Responses	156
8.6	system response	158
8.7	System function	160
9.1	Convolution	165
9.2	Convolution	166
9.3	Autocorrelation	168
9.4	Autocorrelation	169
9.5	Autocorrelation	170
9.6	Autocorrelation	172
9.7	Autocorrelation	173
9.8	Autocorrelation	175
9.9	Autocorrelation	176
9.10	PSD	178
10.1	Cross Correlation	181
10.2	Cross Correlation Sequence	182
10.3	Auto Correlation	183
10.4	Auto Correlation	185
10.5	Auto Correlation	186
10.6	System response	188
11.1	Hilbert Transform	191
11.2	Hilbert Transform	192
11.3	Hilbert Transform	193
11.4	Hilbert Transform	194
11.5	Hilbert Transform	196
11.6	Hilbert Transform	197

11.7 Hilbert Transform	199
11.8 Hilbert Transform	200

Chapter 1

Signals Continuous and Discrete time

Scilab code Exa 1.1 Finding Fundamental Period

```
1 //Scilab Code for Example 1.1 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Determine whether the given signal is periodic or
  not
4 //x(t)=10*(cos(10*pi*t))^2
5 clc;
6 clear;
7 syms t;
8 x=10*(cos(10*%pi*t))^2;
9 disp(x, 'x(t)');
10 t=0:0.01:1;
11 x=10*(cos(10*%pi*t))^2;
12 t=0:0.01:1;
13 plot(t,x, 'r')
14 title('x(t)');
15 xlabel('Time in seconds');
16 disp('the signal is plotted and it shows it is
  periodic');
```

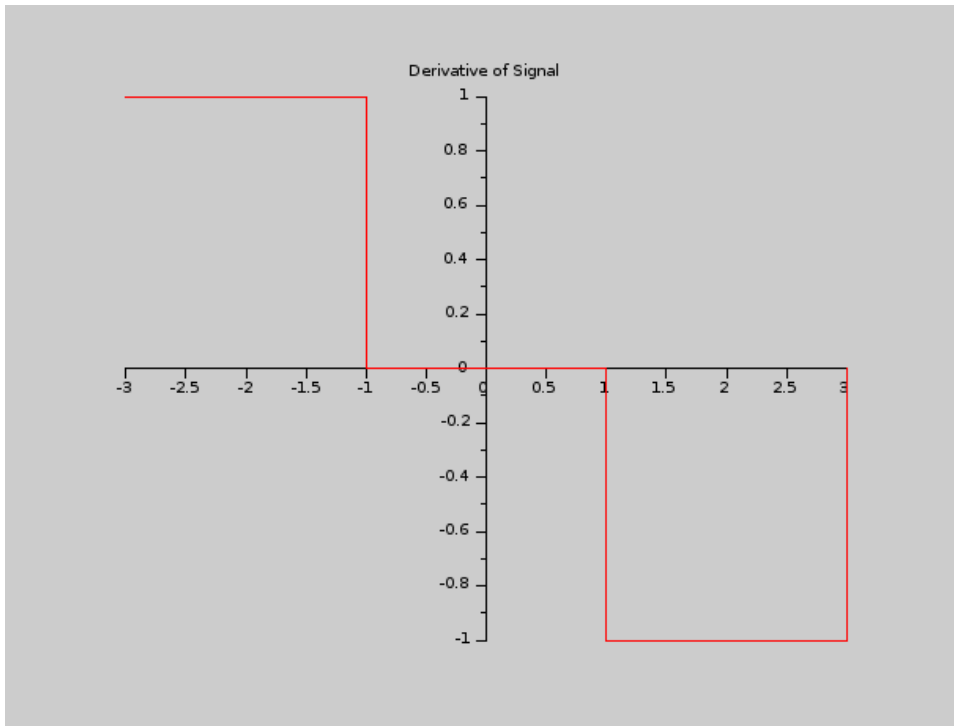


Figure 1.1: Finding Fundamental Period

Scilab code Exa 1.3.a Periodicity

```

1 //Scilab Code for Example 1.3(a) of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //Determine whether the given signal is periodic or
  not
4 //x(t)=3*cos(0.4*pi*t)+2*sin(0.66*t)

```

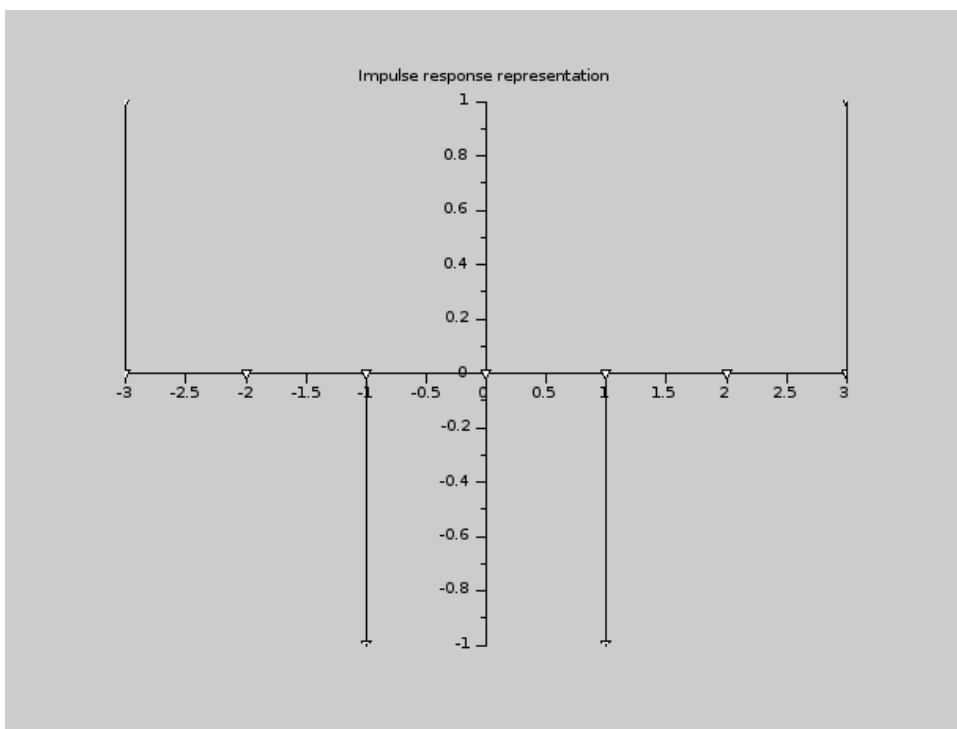


Figure 1.2: Finding Fundamental Period

```

5  clc;
6  clear;
7  syms t;
8  x=3*cos(0.4*%pi*t)+2*sin(0.66*t);
9  disp(x, 'x(t)');
10 t=0:1/50:50;
11 x=3*cos(0.4*%pi*t)+2*sin(0.66*t);
12 t=0:1/50:50;
13 plot(t,x);
14 title('x(t)');
15 xlabel('Time in seconds');
16 disp('plotted the signal and shown that it is not
        periodic and is increasing');

```

Scilab code Exa 1.3.b Periodicity and fundamental period

```

1  //Scilab Code for Example 1.3(b) of Signals and
    systems by
2  //P.Ramakrishna Rao
3  //Determine whether the given signal is periodic or
    not
4  //x(t)=5*cos((4/3)*t)+3*sin(t)
5  clc;
6  clear;
7  syms t;
8  x=5*cos((4/3)*t)+3*sin(t);
9  disp(x, 'x(t)');
10 t=0:1/80:80;
11 x=5*cos((4/3)*t)+3*sin(t);
12 t=0:1/80:80;
13 plot(t,x);
14 title('x(t)');
15 xlabel('Time in seconds');

```

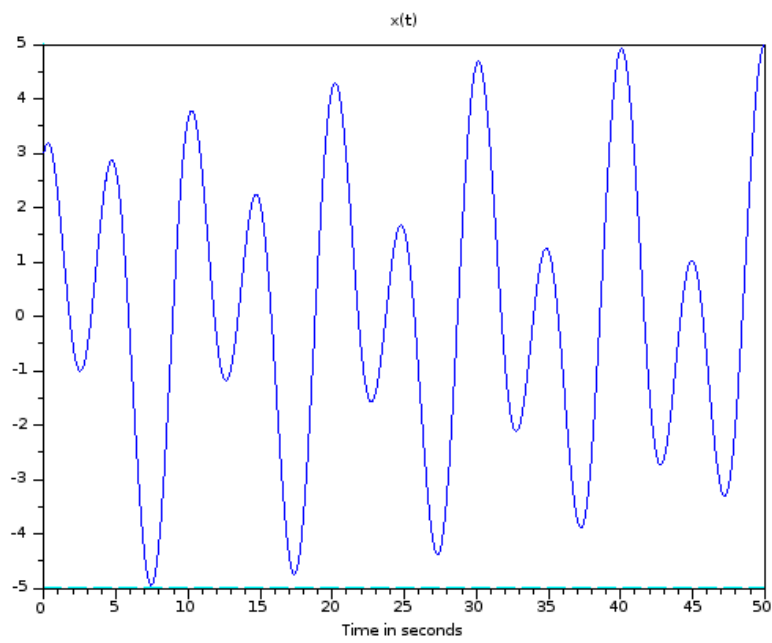



Figure 1.3: Periodicity

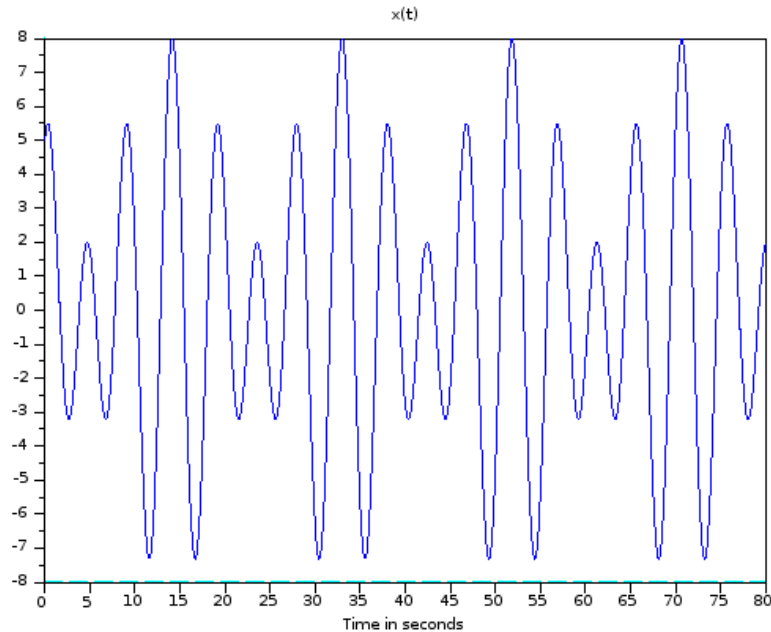


Figure 1.4: Periodicity and fundamental period

16 `disp('plotted the signal and shown that it is
periodic with a period of LCM of 2π and $(2\pi)/(4/3)$ ');`

Scilab code Exa 1.4 Fundamental Period

```

1 //Scilab Code for Example 1.4 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Determine whether the given signal is periodic or
  not

```

```

4 //x(t)=cos(7*t)+sin(4*t)
5 clc;
6 clear;
7 syms t;
8 x=sin(4*t)+cos(7*t);
9 disp(x, 'x(t)');
10 t=0:1/12:12
11 x=sin(4*t)+cos(7*t);
12 t=0:1/12:12;
13 plot(t,x);
14 title('x(t)');
15 xlabel('Time in seconds');
16 disp('plotted the signal and shown that it is
    periodic with period of 2pi');

```

Scilab code Exa 1.5 Fundamental Period

```

1 //Scilab Code for Example 1.5 of Signals and systems
    by
2 //P.Ramakrishna Rao
3 //Determine whether the given signal is periodic or
    not
4 //x(t)=cos(t)+sin(sqrt(2)*t)
5 clc;
6 clear;
7 syms t;
8 x=cos(t)+sin(sqrt(2)*t);
9 disp(x, 'x(t)');
10 for t=0:1:100;
11 x(t+1)=cos(t)+sin(sqrt(2)*t);
12 end
13 t=0:1:100;
14 plot(t,x);

```

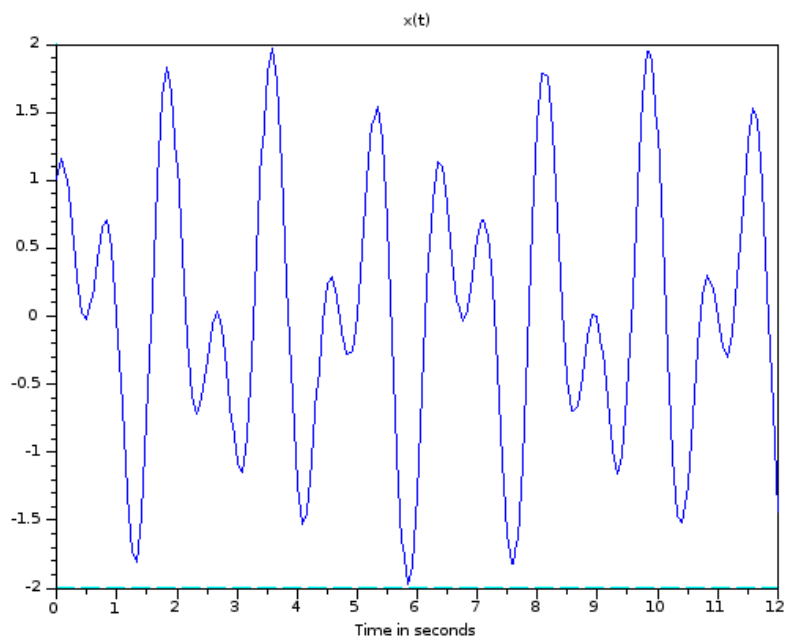


Figure 1.5: Fundamental Period

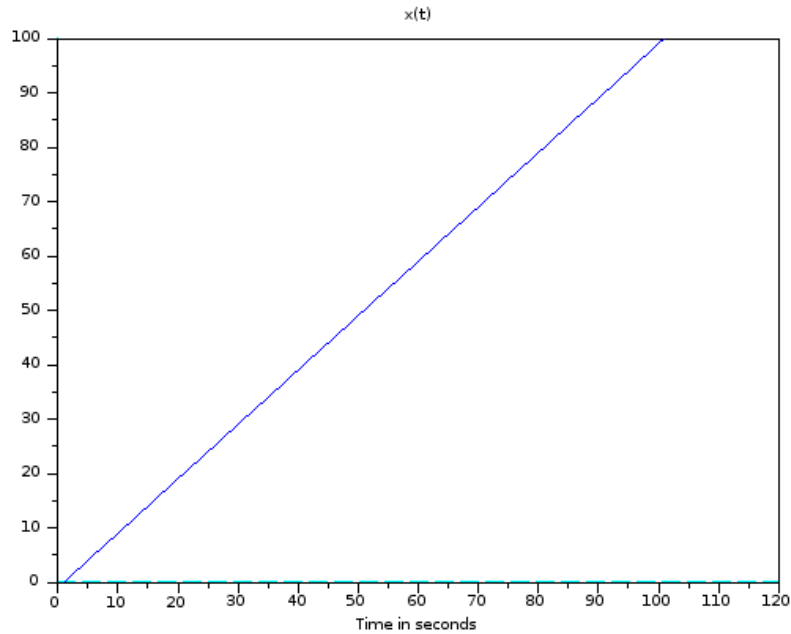


Figure 1.6: Fundamental Period

```

15 title('x(t)');
16 xlabel('Time in seconds');
17 disp('plotted the signal and shown that it is not
      periodic');

```

Scilab code Exa 1.6 Fundamental Period

```

1 //Scilab Code for Example 1.6(i) of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //Determine whether the given signal is periodic or

```

```

not
4  clc;
5  clear;
6
7  n=0:1:10;
8  x(n+1)=2*sin(0.8*pi*n);
9  a=gca();
10 a.x_location="origin";
11 a.y_location="origin";
12 n=0:1:10;
13 plot2d3(n,x,9);
14 title('x(n)');
15 disp('plotting the signal and showing that it is
    periodic with period of 5');

```

Scilab code Exa 1.7 Even and odd Components

```

1 //Scilab Code for Example 1.7 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 n=1;
6 for t=-10:0.1:10;
7     //Function for Even signal
8     y1(n)=0.5*(exp(-t)*u(t)+exp(t)*u(-t));
9     n=n+1;
10 end
11 a=gca();
12 a.x_location="origin";
13 a.y_location="origin";
14 t=-10:0.1:10;
15 //Plot of Even Signal

```

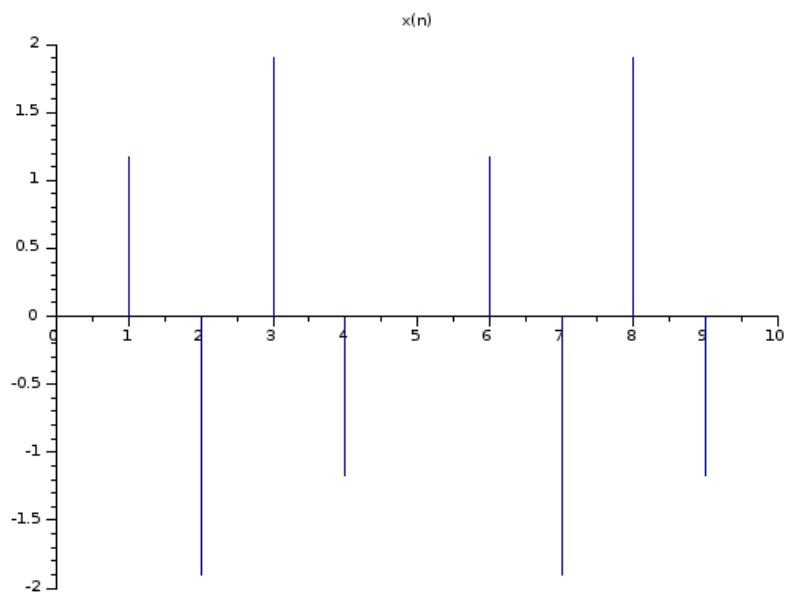


Figure 1.7: Fundamental Period

```

16 plot(t,y1);
17 title('y1(t)');
18 xlabel('Time in seconds');
19 n=1;
20 for t=-1:0.01:1;
21     //Function for Odd signal
22     y2(n)=0.5*(exp(-t)*u(t)-exp(t)*u(-t));
23     n=n+1;
24 end
25 figure(1);
26 a=gca();
27 a.x_location="origin";
28 a.y_location="origin";
29 t=-1:0.01:1;
30 //Plot of Odd Signal
31 plot(t,y2)
32 disp('plotted the signal both in even and odd forms'
      );
33 title('y2(t)');
34 xlabel('Time in seconds');

```

Scilab code Exa 1.11 Waveforms

```

1 //Scilab Code for Example 1.11 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //displaying plots for the given signals
4 clear;
5 clc;
6 for t=-3:1:3
7 y(t+4)=abs(t+3)*u(t+3)-abs(t+1)*u(t+1)-abs(t-1)*u(t

```

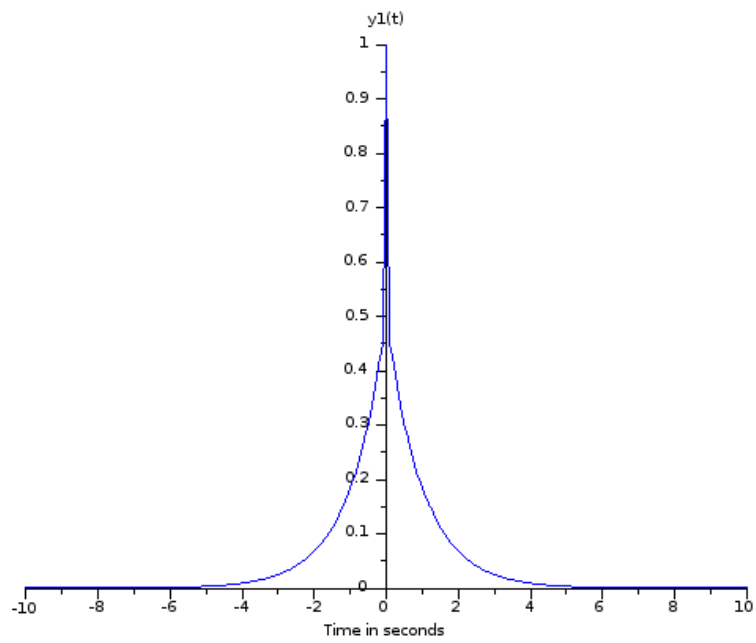



Figure 1.8: Even and odd Components

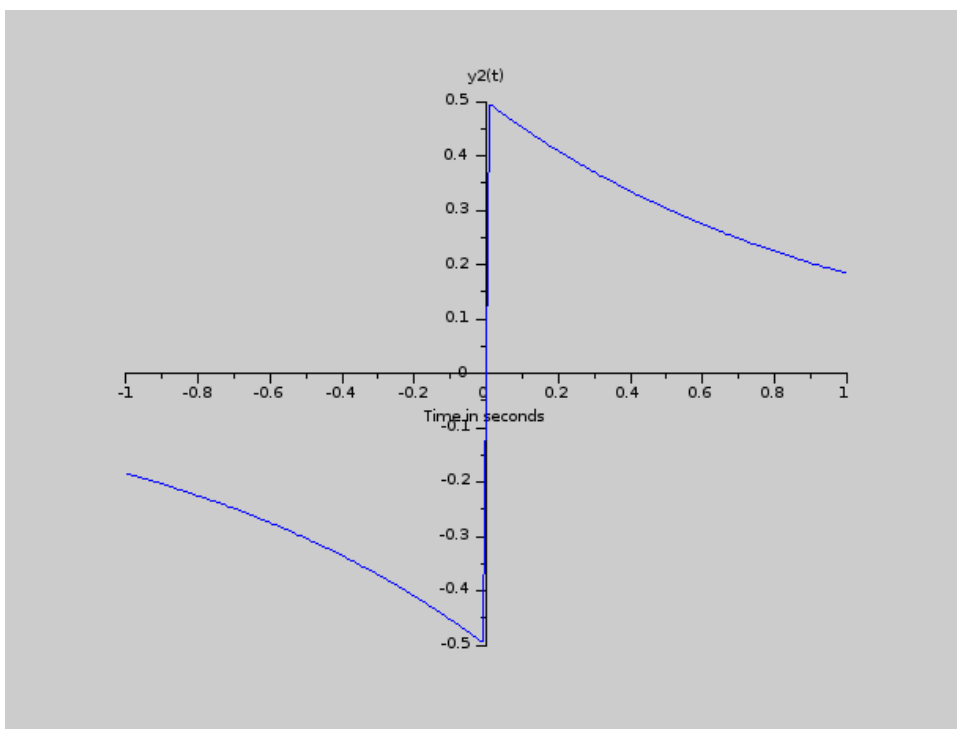


Figure 1.9: Even and odd Components

```

        -1)+abs(t-3)*u(t-3);
8  end
9  t=-3:1:3;
10 //for the main given signal
11 a=gca();
12 a.x_location="origin";
13 a.y_location="origin";
14 plot(t,y);
15 title('Main Signal');
16 dy=0*y;
17 for i=1:6
18 dy(i)=(y(i+1)-y(i))/1;
19 end
20 //for the derivative of the given signal
21 figure(1);
22 a=gca();
23 a.x_location="origin";
24 a.y_location="origin";
25 plot2d2(t,dy);
26 title('Derivative of Signal');
27 dy2=0*dy;
28 dy2(1)=dy(1)-0;
29 for i=1:6
30 dy2(i+1)=(dy(i+1)-dy(i))/1;
31 end
32 //for the impulse response representation or second
    derivative
33 figure(2);
34 a=gca();
35 a.x_location="origin";
36 a.y_location="origin";
37 plot2d3(t,dy2,-5);
38 title('Impulse response representation');

```

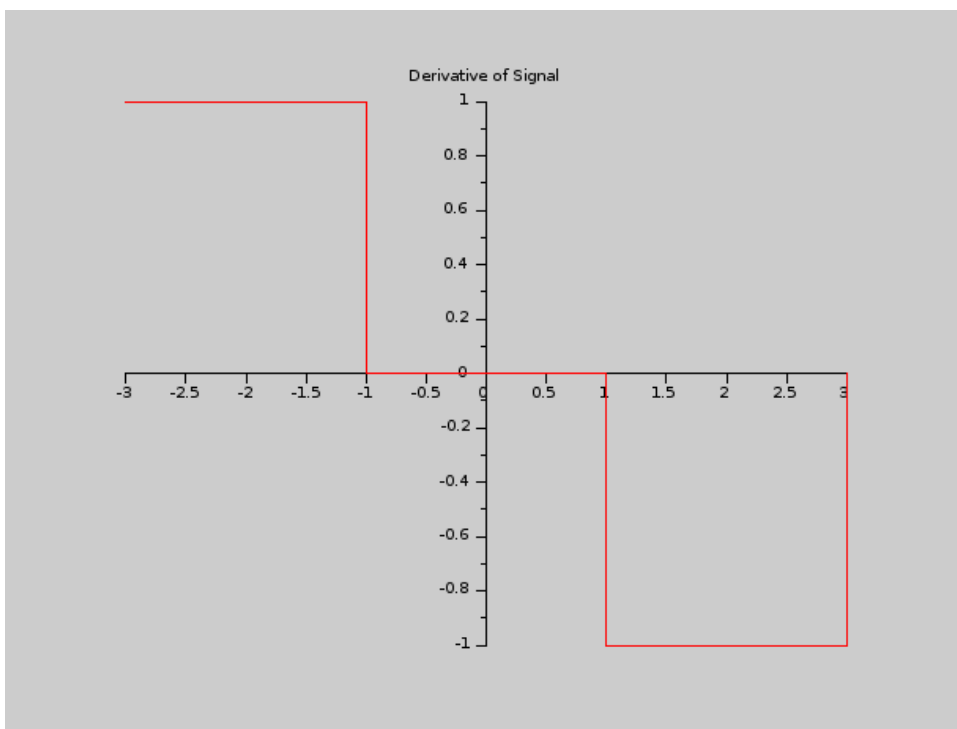


Figure 1.10: Waveforms

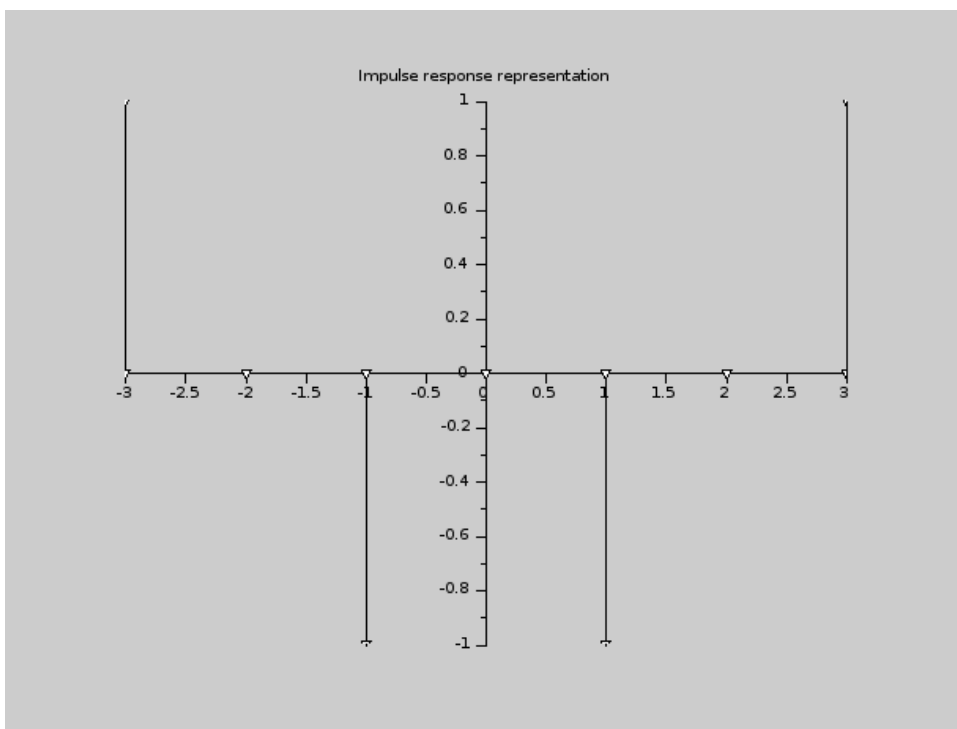


Figure 1.11: Waveforms

Scilab code Exa 1.12 Time Scaling

```
1 //Scilab Code for Example 1.12 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 clc;
4 clear;
5 x=-2:1:3;
6 y=[-1.5,2,2,1,-1.5,2.5];
7 //Plot of x(n)
8 plot2d3(x,y,5);
9 xtitle ( 'Time Scaling x(n)');
10 a = gca(); // get the current axes
11 a.x_location = "origin";
12 a.y_location = "origin";
13 x=-2:1:3;
14 y=[0,-1.5,2,-1.5,0,0];
15 figure(1);
16 //Plot of x(2n)
17 plot2d3(x,y,5);
18 a = gca(); // get the current axes
19 a.x_location = "origin";
20 a.y_location = "origin";
21 xtitle ( 'Time Scaling x(2n)' );
```

Scilab code Exa 1.13 Plot at a particular sampling frequency

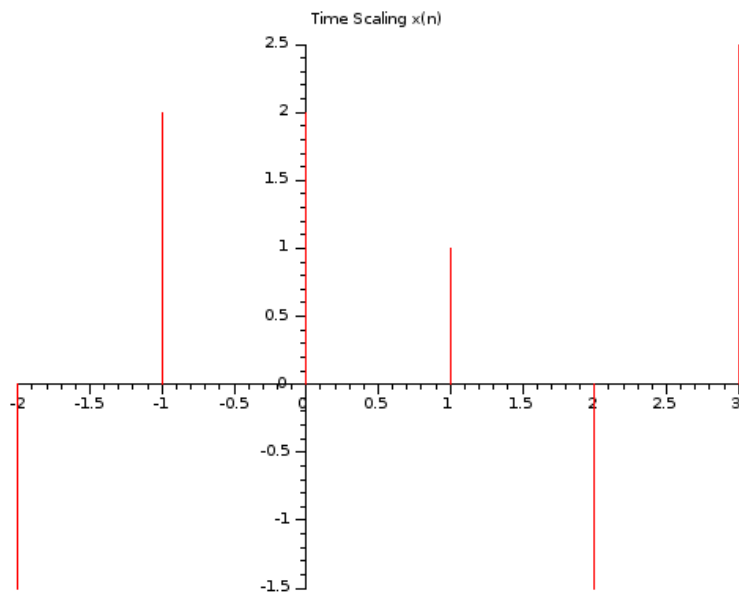


Figure 1.12: Time Scaling

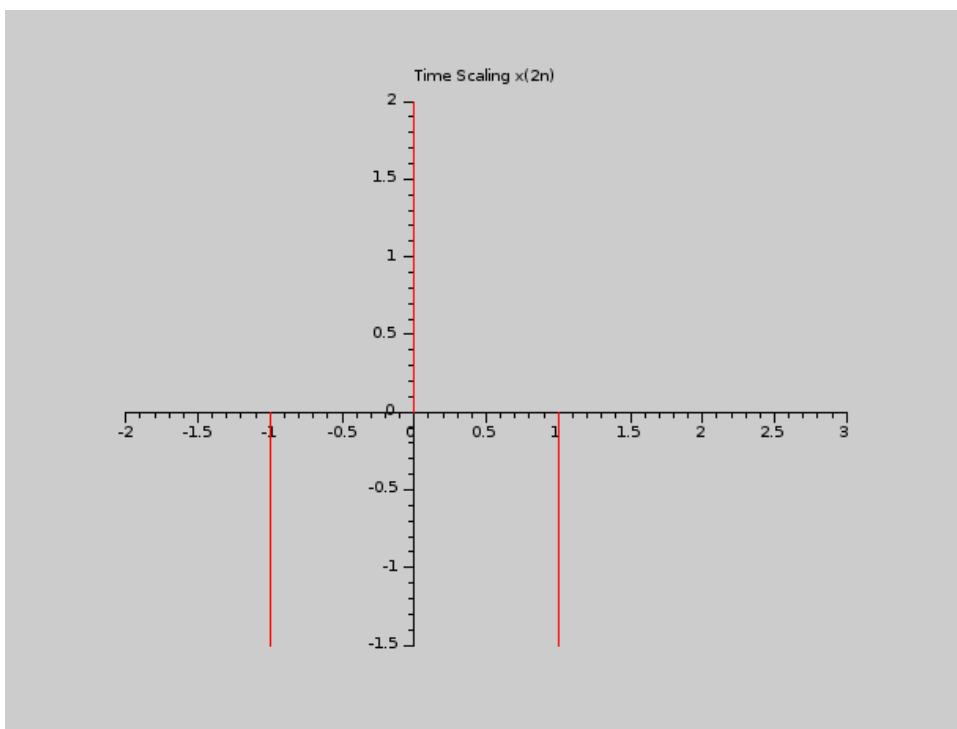


Figure 1.13: Time Scaling


```
1 //Scilab Code for Example 1.13 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 clc;
4 clear;
5 //Discrete-time sequence plot
6 clc
7 A=10;
8 t=0:1/1000:1;
9 x=A*sin(2*%pi*100*t);
10 q=plot2d3(t,x);
11 disp('displaying a function plot on discrete time
       scale that has a sampling frequency of 1000
       samples ps');
```

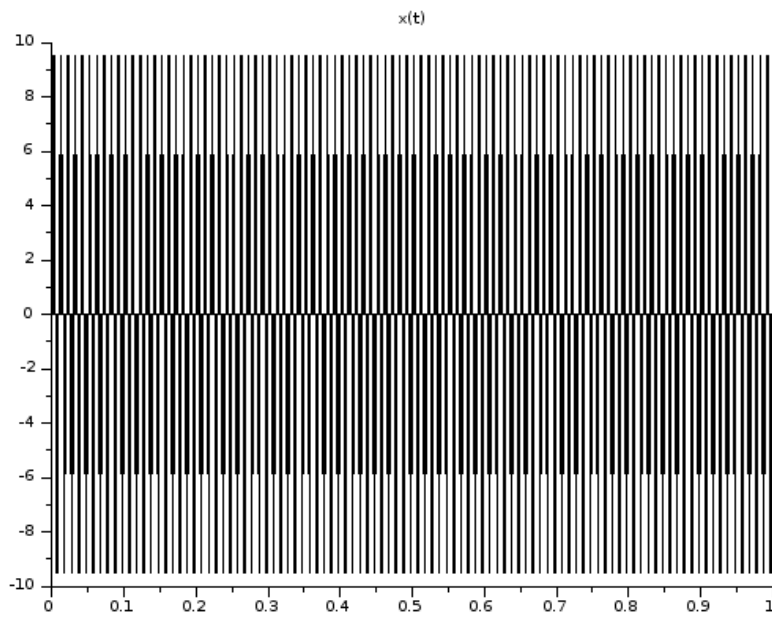


Figure 1.14: Plot at a particular sampling frequency

Chapter 2

Laplace and Z Transform

Scilab code Exa 2.1 Laplace Transform

```
1 //Scilab Code for Example 2.1 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Laplace Transform
4 clear;
5 clc;
6 syms t s X
7 X=laplace(exp(-3*t)*cos(2*pi*100*t),t,s)
8 disp('On Simplification ')
9 //After Simplifying the above equation
10 X=(s+3)/((s+3)^2 + (200*pi)^2);
11 disp("Re(s)>-3",X,"X(s)");
12 //Re(s)>-3
```

Scilab code Exa 2.2 Inverse Laplace

```
1 //Scilab Code for Example 2.2 of Signals and systems
  by
```

```

2 //P.Ramakrishna Rao
3 //Inverse Laplace Transform
4 clc;
5 clear;
6 syms t s
7 X=(s+2)/(s^2+8*s+25)
8 f4=ilaplace(X);
9 disp(f4*'u(t)', 'x(t)=');
10 //t>=0

```

Scilab code Exa 2.3 Initial Value Theorem

```

1 //Scilab Code for Example 2.3 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms s0;
6 s=%s;
7 I=(s+8)/(s^2+6*s+13)
8 i=pfss(s*I)
9 disp(i(1), 'sF(s)(1)=')
10 disp(i(2), 'sF(s)(2)=')
11 I1=(2*s0-13)/(s0^2+6*s0+13);
12 I2=1;
13 Io1=limit(I1,s0,10^8);
14 Io2=limit(I2,s0,10^8);
15 Ix=2-((25*s0+26)/(s0^2+6*s0+13));
16 f0=(Io1)+(Io2);
17 f0_dash=limit(Ix,s0,10^8);
18 disp(f0, 'INITIAL VALUE OF f(t) i.e. f(0)=');
19 disp(abs(f0_dash), 'INITIAL VALUE OF f(t) i.e. f''(0)
  =');

```

Scilab code Exa 2.4 Laplace Transform

```
1 //Scilab Code for Example 2.4 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 //Time shifted laplace Transform
6 syms t y s a0;
7 y=laplace('t*exp(-s*a0)',t,s);
8 disp("Re(s)>0",y,"X(s)");
```

Scilab code Exa 2.6 Circuit Current

```
1 //Scilab Code for Example 2.6 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms s X x t R C V Vo;
6 //After solving for I(s)
7 //I(s)=(V-Vo)/R . 1/(s+1/RC)
8 X=(V-Vo)/((s+1/(R*C))*R);
9 disp(X,"I(s)=");
10 x=ilaplace(X);
11 disp(x,"i(t)=");
```

Scilab code Exa 2.7 Inverse Laplace Transform

```

1 //Scilab Code for Example 2.7 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Unilateral Laplace Transform using partial
  fraction
4 clear;
5 clc;
6 syms t s;
7 s= %s;
8 a1=pfss((s+3)/(s^2+3*s+2))
9 f1=ilaplace(a1(1))
10 f2=ilaplace(a1(2))
11 fy=f1+f2
12 disp(fy*'u(t)', 'i) f(t)=')
13 a2=pfss((2*s-1)/(s^2+2*s+1))
14 a2(1)=2/(s+1)
15 a2(2)=-3/(s+1)^2
16 f1=ilaplace(a2(1))
17 f2=ilaplace(a2(2))
18 fz=f1+f2
19 disp(fz*'u(t)', 'ii) f(t)=')

```

Scilab code Exa 2.8 Inverse Laplace Transform

```

1 //Scilab Code for Example 2.8 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Unilateral Laplace Transform
4 clear;
5 clc;
6 syms t s
7 s=%s;
8 a=ilaplace(1/(s^3+s^2))
9 b=a-3;
10 disp(a*'u(t)'+b*'u(t-3)', 'x(t)=')

```

Scilab code Exa 2.9 Circuit Current

```
1 //Scilab Code for Example 2.2 of Signals and systems
   by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms s X;
6 //After solving for I(s)
7 //I(s)=2/(s+1)
8 X=2/(s+1)
9 x=ilaplace(X);
10 disp(x," i (t)=")
```

Scilab code Exa 2.10 Laplace Transform

```
1 //Scilab Code for Example 2.10 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //The value of X(s) is found by solving the
   differential equation
4 clear;
5 clc;
6 syms t s;
7 s= %s;
8 X=pfss((s^2+8*s+6)/((s+2)*(s+3)*s));
9 X(1)=1/s;
10 f1=ilaplace(X(1))
11 f2=ilaplace(X(2))
12 f3=ilaplace(X(3))
13 fz=f1+f2+f3;
14 disp(fz* 'u(t) ', "c) x(t)=");
```

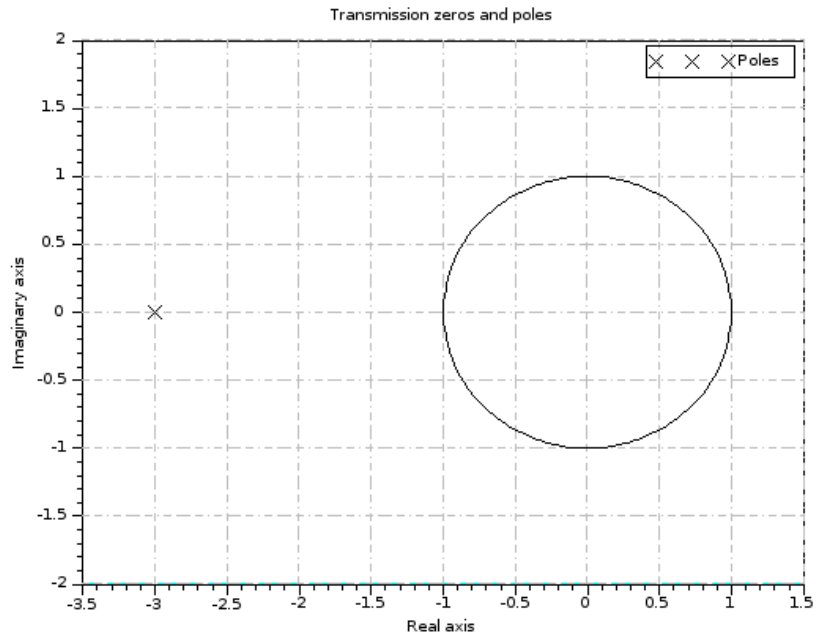


Figure 2.1: Laplace Transform

Scilab code Exa 2.15 Laplace Transform

```

1 //Scilab Code for Example 2.15 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms t y;
6 s=%s;
7 y=laplace(5*exp(-3*t),t,s);

```



```
8 disp(y,"X(s)=");
9 disp("Re(s)>-3");
10 y=5/(s+3);
11 plzr(y);
```

Scilab code Exa 2.16 Laplace Transform

```
1 //Scilab Code for Example 2.16 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms t y;
6 s=%s;
7 y=laplace(2*exp(-2*t)+3*exp(-3*t),t,s);
8 disp(y,"X(s)=");
9 y=(2/(s+2))+(3/(s+3));
10 plzr(y);
```

Scilab code Exa 2.17 Laplace Transform

```
1 //Scilab Code for Example 2.17 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms t y;
6 s=%s;
7 y=laplace(-2*exp(2*t)-3*exp(3*t),t,s);
8 disp(y,"X(s)=");
9 disp("Re(s)<2");
```

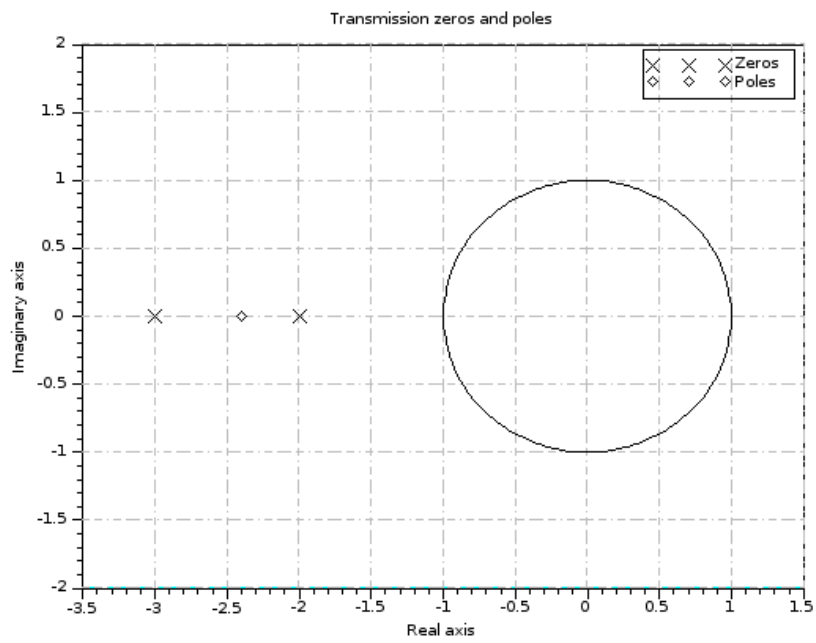


Figure 2.2: Laplace Transform

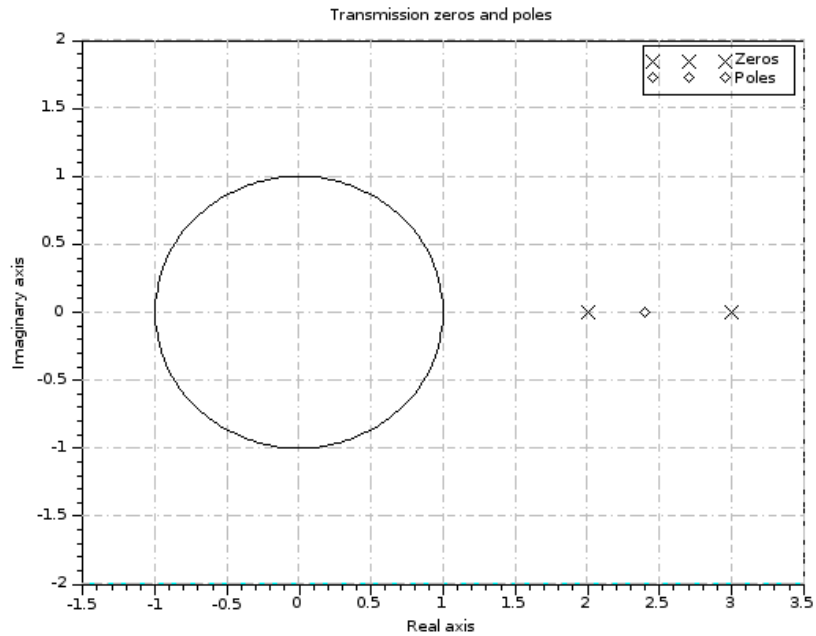


Figure 2.3: Laplace Transform

```
10 y=(-2/(s-2))+(-3/(s-3));
11 plzr(y);
```

Scilab code Exa 2.19 Laplace Transform

```
1 //Scilab Code for Example 2.19 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms t y;
```

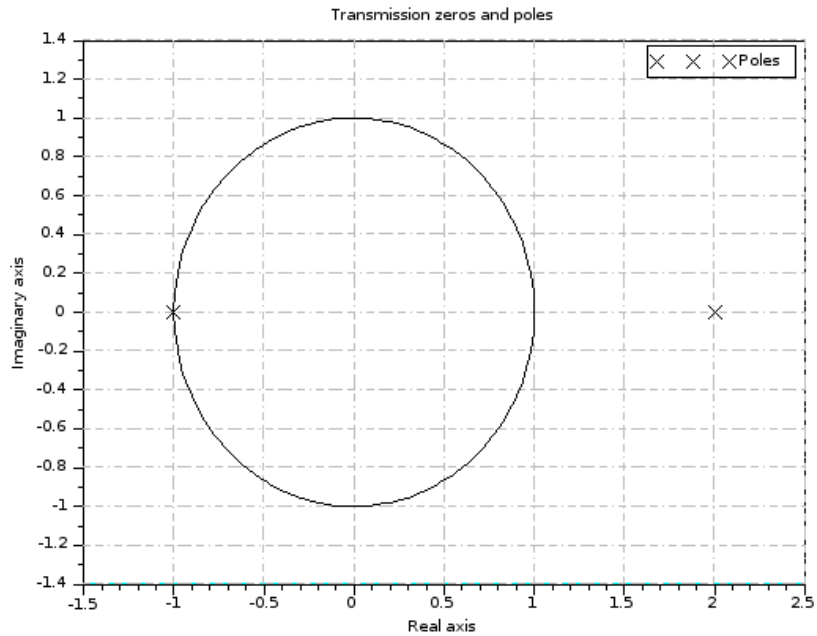


Figure 2.4: Laplace Transform

```

6 s=%s;
7 y=laplace(exp(-t)-exp(2*t),t,s);
8 disp(y,"X(s)=");
9 y=(1/(s+1))-(1/(s-2));
10 plzr(y);

```

Scilab code Exa 2.21 Unilateral Z Transform

```

1 //Scilab Code for Example 2.21 of Signals and
  systems by
2 //P.Ramakrishna Rao

```

```

3 //Z- transform of a^n u(n)
4 clear;
5 clc ;
6 close ;
7 syms a n z;
8 x1 =1/2;
9 x2=1/3;
10 X1= symsum (x1*(z^(-n)),n ,0, %inf );
11 X2= symsum (x2*(z^(-n)),n ,0, %inf );
12 X=X1+X2;
13 //Display the result
14 disp (X,"Z-transform of u(n) is:");
15 disp('ROC is the Region |z|> 1/2');

```

Scilab code Exa 2.29 Z Transform

```

1 //Scilab Code for Example 2.29 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Z- transform of a^n u(n)
4 clear;
5 clc ;
6 close ;
7 syms a n z;
8 x =1;
9 X= symsum (x*(z^(-n)),n ,0, %inf );
10 //Display the result
11 disp (X,"Z-transform of u(n) is:");
12 disp('ROC is the Region |z|> 1')

```

Scilab code Exa 2.37 Inverse Z Transform

```

1 //Scilab Code for Example 2.37 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //Inverse Z Transform:ROC  $1 < |z| < 2$ 
4 clear;
5 clc;
6 z = %z;
7 syms n z1;
8 //To find out Inverse z transform z must be linear z
  = z1
9 X = (z*(z^2-z+1))/((z-0.5)*(z-2)*(z-1))
10 X1 = denom(X);
11 zp = roots(X1);
12 X1 = (z1*(z1^2-z1+1))/((z1-0.5)*(z1-2)*(z1-1))
13 F1 = X1*(z1^(n-1))*(z1-zp(1))
14 F2 = X1*(z1^(n-1))*(z1-zp(2))
15 F3 = X1*(z1^(n-1))*(z1-zp(3))
16 h1 = limit(F1,z1,zp(1));
17 disp(h1*'u(-n-1)', 'h1[n]=')
18 h2 = limit(F2,z1,zp(2));
19 disp((h2)*'u(n)', 'h2[n]=')
20 h3 = limit(F3,z1,zp(3));
21 disp((h3)*'u(n)', 'h3[n]=')
22 disp((h3)*'u(n)'+(h2)*'u(n)'-(h1)*'u(-n-1)', 'h[n]=')
23 ////Result
24 // h[n]=1* 0.5 ^n *u(n) - 2*u(n) - 2* 2^n *u(- n -
  1)

```

Scilab code Exa 2.38.a Inverse Z Transform

```

1 //Scilab Code for Example 2.38(i) of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //Inverse Z Transform:ROC  $|z| > 2$ 
4 clear;

```

```

5  clc;
6  z = %z;
7  syms n z1;
8  //To find out Inverse z transform z must be linear z
   = z1
9  X  =z^2/(z^2+3*z+2);
10 X1 = denom(X);
11 zp = roots(X1)
12 X1 = z1^2/(z1^2+3*z1+2);
13 F1 = X1*(z1^(n-1))*(z1-zp(1));
14 F2 = X1*(z1^(n-1))*(z1-zp(2));
15 h1 = limit(F1,z1,zp(1));
16 disp(h1*'u(n)', 'h1[n]=');
17 h2 = limit(F2,z1,zp(2));
18 disp((h2)*'u(n)', 'h2[n]=');
19 disp((h1)*'u(n)'+(h2)*'u(n)', 'h[n]=');
20 ////Result
21 //  h[n]= (2(- 2)^n+1 - (- 1)^n)*u(n)

```

Scilab code Exa 2.38.b Inverse Z Transform

```

1  //Scilab Code for Example 2.38(ii) of Signals and
   systems by
2  //P.Ramakrishna Rao
3  //Inverse Z Transform:ROC |z|>2
4  clear;
5  clc;
6  z = %z;
7  syms n z1;
8  //To find out Inverse z transform z must be linear z
   = z1
9  X  =(z+1)/(z^2+5*z+4)
10 X1 = denom(X);
11 zp = roots(X1)
12 X1 = 1/(z1+4);

```

```

13 F1 = X1*(z1^(n))*(z1-zp(1));
14 h1 = limit(F1,z1,zp(1))
15 disp(-(h1)*'u(-n-1)', 'h[n]= ');
16 ///Result
17 // h[n]= (- (- 4)^n)*u(-n-1)

```

Scilab code Exa 2.39 Inverse Z Transform

```

1 //Scilab Code for Example 2.39 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //Inverse Z Transform:ROC |z|>2
4 clear;
5 clc;
6 z = %z;
7 syms n z1;
8 //To find out Inverse z transform z must be linear z
  = z1
9 X =2/(z^-1+2);
10 X1 = denom(X);
11 zp = roots(X1)
12 X1 =z1/(z1+0.5);
13 F1 = X1*(z1^(n-1))*(z1-zp(1));
14 h1 = limit(F1,z1,zp(1))
15 disp(-(h1)*'u(-n-1)', 'h[n]= ');
16 ///Result
17 // h[n]= (- (- 0.5)^n)*u(-n-1)

```

Scilab code Exa 2.40 Inverse Z Transform

```

1 //Scilab Code for Example 2.40 of Signals and
  systems by
2 //P.Ramakrishna Rao

```



```

3 //Inverse Z Transform:ROC |z|>2
4 clear;
5 clc;
6 z = %z;
7 syms n z1;
8 //To find out Inverse z transform z must be linear z
   = z1
9 X =z^2/((z-0.5)*(z-1));
10 X1 = denom(X);
11 zp = roots(X1)
12 X1 = z1^2/((z1-0.5)*(z1-1));
13 F1 = X1*(z1^(n-1))*(z1-zp(1));
14 F2 = X1*(z1^(n-1))*(z1-zp(2));
15 h1 = limit(F1,z1,zp(1));
16 disp(h1*'u(n)', 'h1[n]=')
17 h2 = limit(F2,z1,zp(2));
18 disp((h2)*'u(n)', 'h2[n]=');
19 disp(-(h1)*'u(-n-1)'-(h2)*'u(-n-1)', 'h[n]=');
20 ////Result
21 // h[n]= ((0.5)^n - 2)*u(-n-1)

```

Chapter 3

Fourier Series of Continuous Time Signals

Scilab code Exa 3.13 Complex Fourier Exponential Series

```
1 //Scilab Code for Example 3.13 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Complex Exponential Fourier Expansion
4 //A=3.14 or pi
5 clear;
6 close;
7 clc;
8 T0=4;
9 t=-5.99:0.01:6;
10 t_temp=0.01:0.01:T0/2;
11 s=length(t)/length(t_temp);
12 x=[];
13 for i=1:s
14     if modulo(i,2)==1 then
```

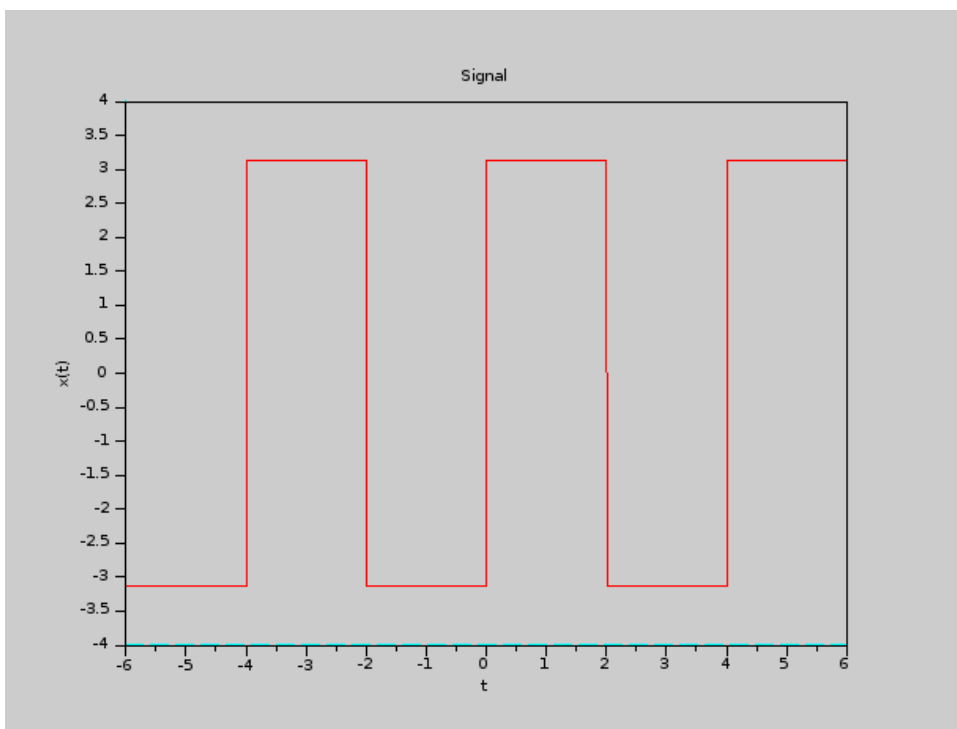


Figure 3.1: Complex Fourier Exponential Series

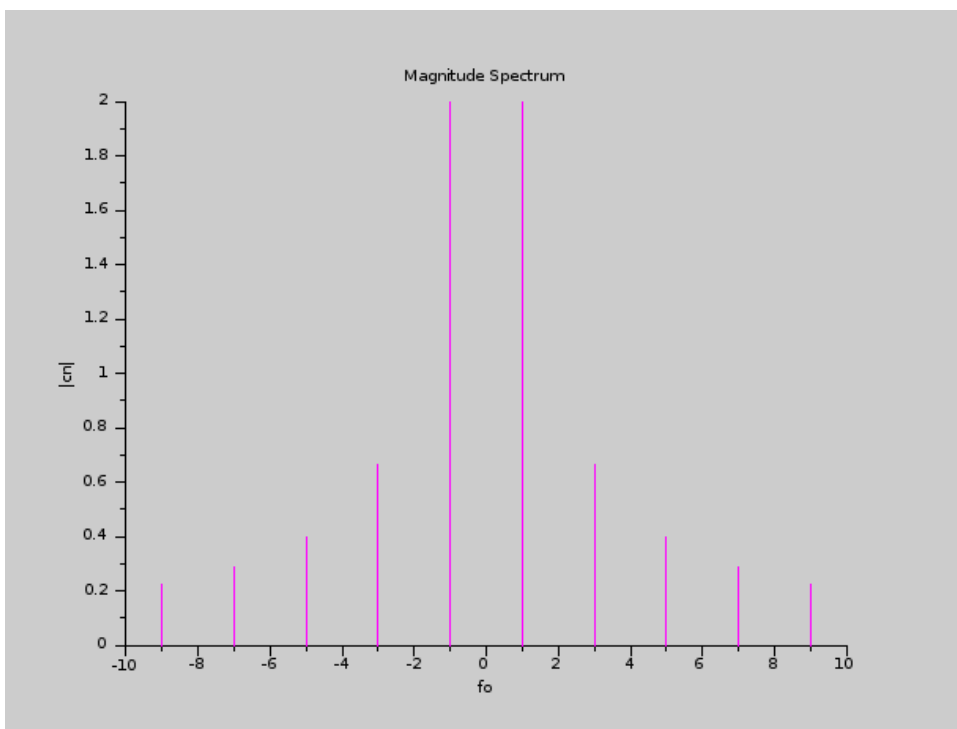


Figure 3.2: Complex Fourier Exponential Series

```

15         x=[x -ones(1,length(t_temp))*%pi];
16     else
17         x=[x ones(1,length(t_temp))*%pi];
18     end
19 end
20 figure(1);
21 title('Signal');
22 ylabel('x(t)');
23 xlabel('t');
24 plot(t,x,'r');
25 w0=%pi/2;
26 for k=-10:10
27     cc(k+11,:)=exp(-%i*k*w0*t);
28     ck(k+11)=x*cc(k+11,:)/length(t);
29     if abs(ck(k+11))<0.01 then
30         ck(k+11)=0;
31     else if real(ck(k+11))<0.1 then
32         ck(k+11)=%i*imag(ck(k+11));
33     end
34     end
35     if k==0 then
36         c0=ck(k+11);
37     end
38 end
39 q=abs(ck);
40 figure(2);
41 title('Magnitude Spectrum');
42 ylabel('|cn|');
43 xlabel('fo');
44 f=-10:1:10;
45 plot2d3(f,q)

```

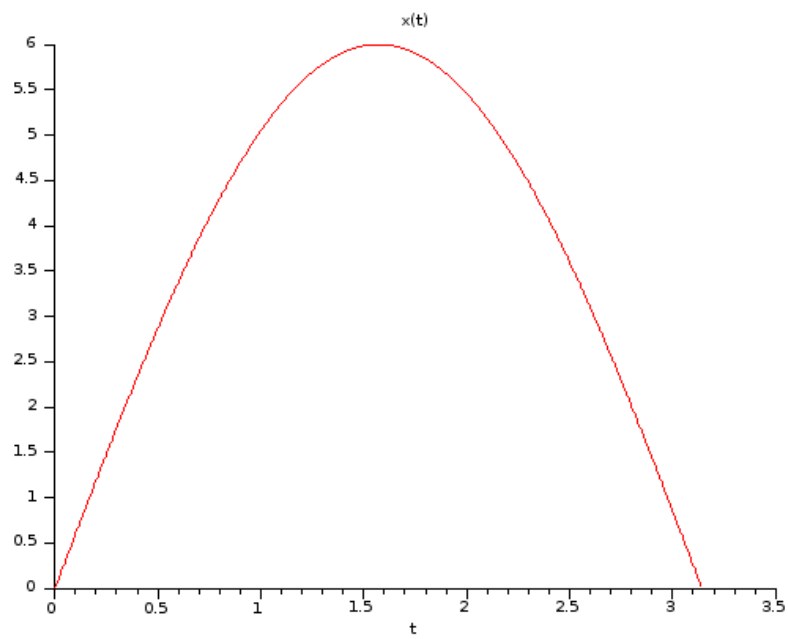


Figure 3.3: Complex Fourier Exponential Series

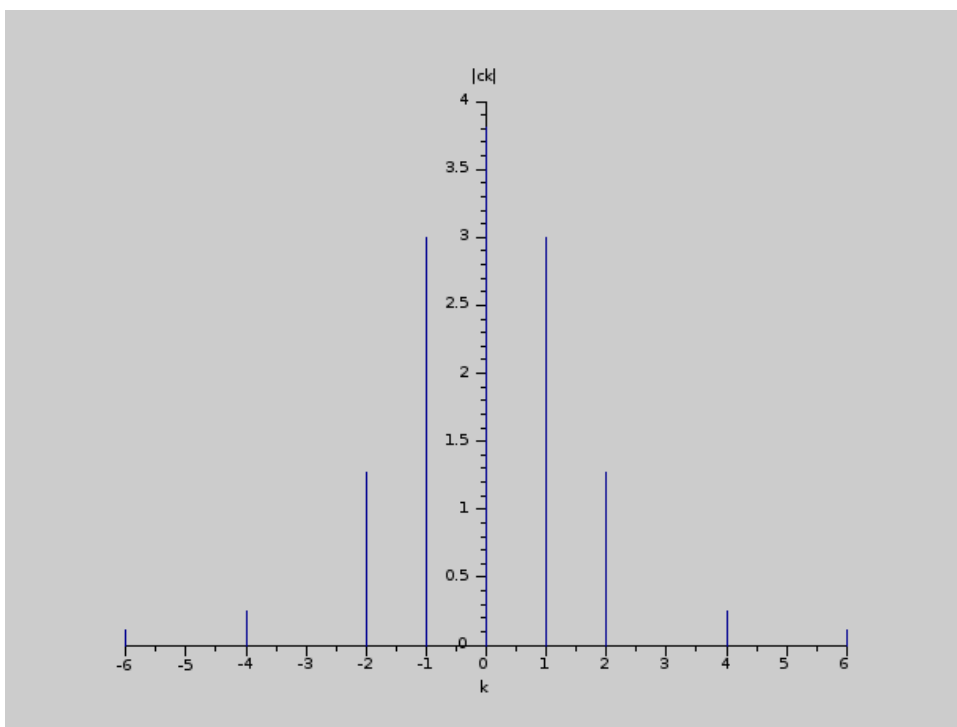


Figure 3.4: Complex Fourier Exponential Series

Scilab code Exa 3.14 Complex Fourier Exponential Series

```
1 //Scilab Code for Example 3.14 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //CTFS coefficients of a periodic signal
4 //x(wt) = A sin wt, 0<wt<pi, and 0, pi<|t|<2*pi
5 clear;
6 clc;
7 A=6;
8 T =2*%pi;
9 T1 = T/2;
10 t = 0:0.01:T1;
11 Wo = 2*%pi/T;
12 xt =A*sin(Wo*t);
13 for k =0:6
14     C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
15     c(k+1) = xt*C(k+1,:)/length(t);
16     if(abs(c(k+1))<=0.01)
17         c(k+1)=0;
18     end
19 end
20 c =c';
21 c_conj = real(c(:))-sqrt(-1)*imag(c(:));
22 ck = [c_conj($:-1:1)',c(2:$)];
23 k = 0:6;
24 k = [-k($:-1:1),k(2:$)];
25 figure(1);
26 c = gca();
27 c.y_location = "origin";
28 c.x_location = "origin";
29 //c.data_bounds=[-2,0;2,2];
30 plot2d(t,xt,5)
31 poly1 = c.children(1).children(1);
32 poly1.thickness = 3;
33 title('x(t)')
34 xlabel('t')
35 figure(2);
```

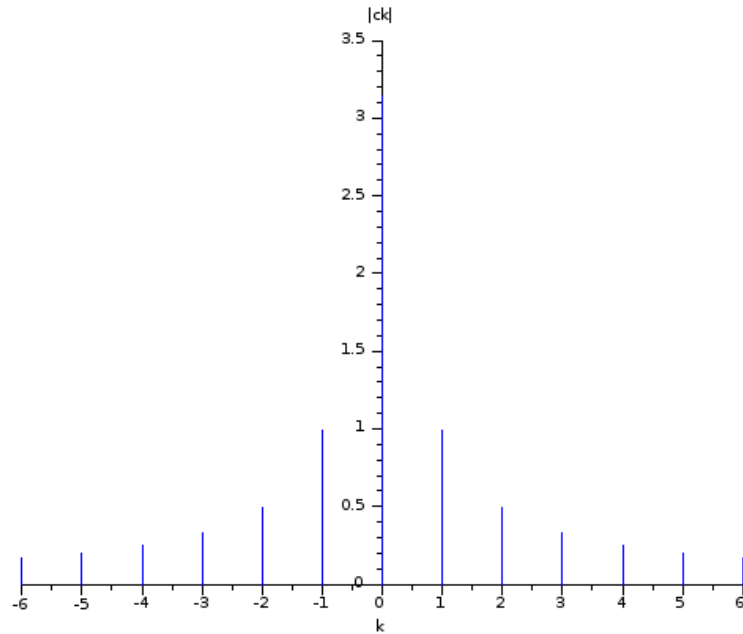



Figure 3.5: Complex Fourier Exponential Series

```

36 c = gca();
37 c.y_location = "origin";
38 c.x_location = "origin";
39 plot2d3('gnn',k,abs(ck),9)
40 poly1 = c.children(1).children(1);
41 poly1.thickness = 3;
42 title('|ck|')
43 xlabel('k')

```

Scilab code Exa 3.16 Complex Fourier Exponential Series

```

1 //Scilab Code for Example 3.16 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //A=%pi or 3.14
4 clear;
5 clc;
6 //Trigonometric Fourier Coefficients
7 for n=0:5
8 a(n+1)=integrate('t*cos(2*%pi*n*t)', 't', 0, 1);
9 end
10 for n=0:5
11 b(n+1)=integrate('t*sin(2*%pi*n*t)', 't', 0, 1);
12 end
13 disp(%pi*a(1), "an(a0)")
14 disp("an(a1—>a5)")
15 for n=1:5
16 disp(2*a(n+1)*%pi)
17 end
18 disp("bn(b1—>b5)")
19 for n=1:5
20 disp(2*%pi*b(n+1))
21 end
22 //CTFS coefficients of a periodic signal
23 //x(t) =t
24 t = 0:0.01:1;
25 xt =2*%pi*t;
26 //
27 for k =0:6
28     C(k+1,:) = exp(-sqrt(-1)*2*%pi*t*k);
29     c(k+1) = xt*C(k+1,:)'/length(t);
30     if(abs(c(k+1))<=0.01)
31         c(k+1)=0;
32     end
33 end
34 c =c';
35 c_conj = real(c(:))-sqrt(-1)*imag(c(:));
36 ck = [c_conj($:-1:1)', c(2:$)];
37 k = 0:6;

```

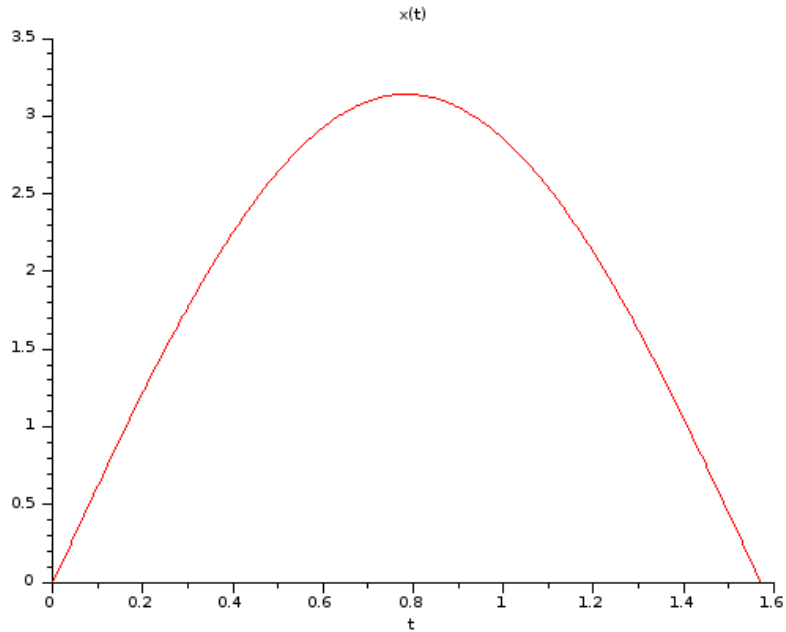


Figure 3.6: Complex Fourier Exponential Series

```

38 k = [-k($:-1:1),k(2:$)];
39 c = gca();
40 c.y_location = "origin";
41 c.x_location = "origin";
42 plot2d3('gnn',k,abs(ck))
43 poly1 = c.children(1).children(1);
44 poly1.thickness = 3;
45 title('|ck|')
46 xlabel('k')

```

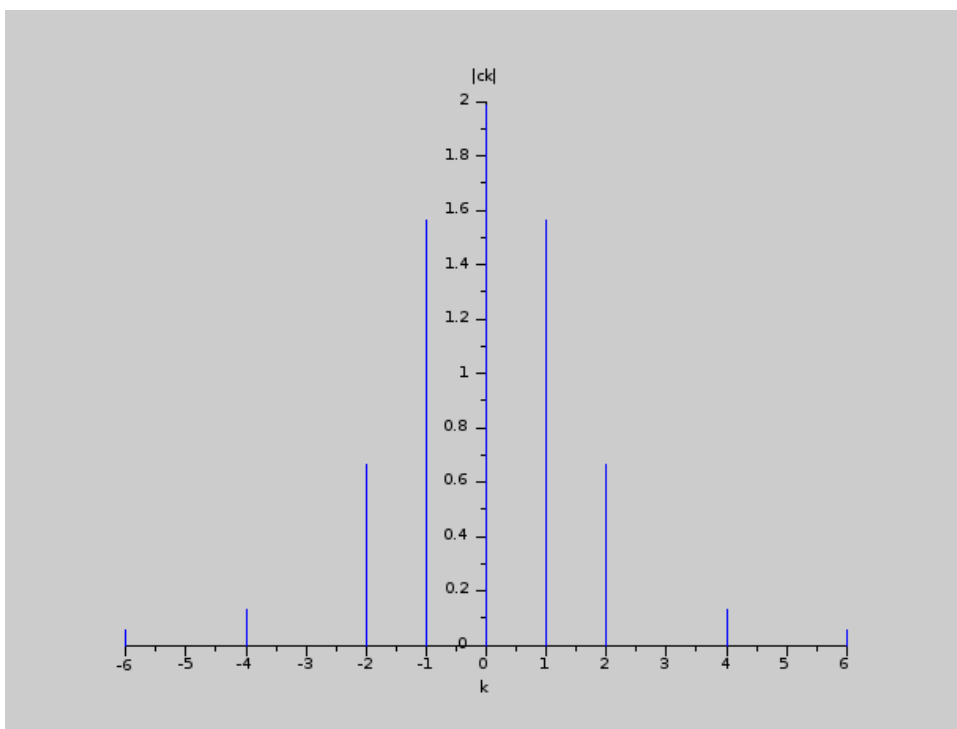


Figure 3.7: Complex Fourier Exponential Series

Scilab code Exa 3.17 Complex Fourier Exponential Series

```
1 //Scilab Code for Example 3.17 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //CTFS coefficients of a periodic signal
4 //x(wt) = Asin wt, 0<wt<pi
5 clear;
6 close;
7 clc;
8 A=3.14;
9 T =%pi;
10 T1 = T/2;
11 t = 0:0.01:T1;
12 Wo = 2*%pi/T;
13 xt =A*sin(Wo*t);
14 //
15 for k =0:6
16     C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
17     c(k+1) = xt*C(k+1,:)'/length(t);
18     if(abs(c(k+1))<=0.01)
19         c(k+1)=0;
20     end
21 end
22 c =c';
23 c_conj = real(c(:))-sqrt(-1)*imag(c(:));
24 ck = [c_conj($:-1:1)',c(2:$)];
25 k = 0:6;
26 k = [-k($:-1:1),k(2:$)];
27 c = gca();
28 c.y_location = "origin";
29 c.x_location = "origin";
30 //c.data_bounds=[-2,0;2,2];
31 plot2d(t,xt,5)
32 poly1 = c.children(1).children(1);
33 poly1.thickness = 3;
34 title('x(t)')
35 xlabel('t')
```

```

36 figure(1);
37 c = gca();
38 c.y_location = "origin";
39 c.x_location = "origin";
40 plot2d3('gmn',k,abs(ck))
41 poly1 = c.children(1).children(1);
42 poly1.thickness = 3;
43 title('|ck|')
44 xlabel('k')

```

Scilab code Exa 3.18 Trigonometric Series

```

1 //Scilab Code for Example 3.18 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //A=%pi or 3.14
4 clear;
5 clc;
6 //Trigonometric Fourier Coefficients
7 a(1)=integrate('sin(w)', 'w', 0, %pi);
8 for n=1:8
9 a(2*n+1)=integrate('sin(w+2*n*w)', 'w', 0, %pi)+
   integrate('sin(w-2*w*n)', 'w', 0, %pi);
10 end
11 for n=0:8
12 b(n+1)=0;
13 end
14 disp(abs(a(1)), "an(a0)");
15 disp("an(a1—>a8)");
16 n=1:8;
17 disp(2*a(n+1));
18 disp("bn(b1—>b8)");
19 n=1:8;
20 disp(b(n));

```

Chapter 4

The Continuous Time Fourier Transform

Scilab code Exa 4.1 Magnitude and Phase Spectra

```
1 //Scilab Code for Example 4.1 of Signals and systems
   by
2 //P.Ramakrishna Rao
3 //Plotting Magnitude and Phase spectrum
4 clc;
5 clear;
6 A=8;
7 Dt=0.005;
8 T1=4;
9 t=-T1/2:Dt:T1/2;
10 q=length(t)
11 for i=-(q/2)+1:q/2
12     if i>-q/4 & i<q/4 then
13         xt(i+(q/2))=A;
14     else xt(i+(q/2))=0;
```

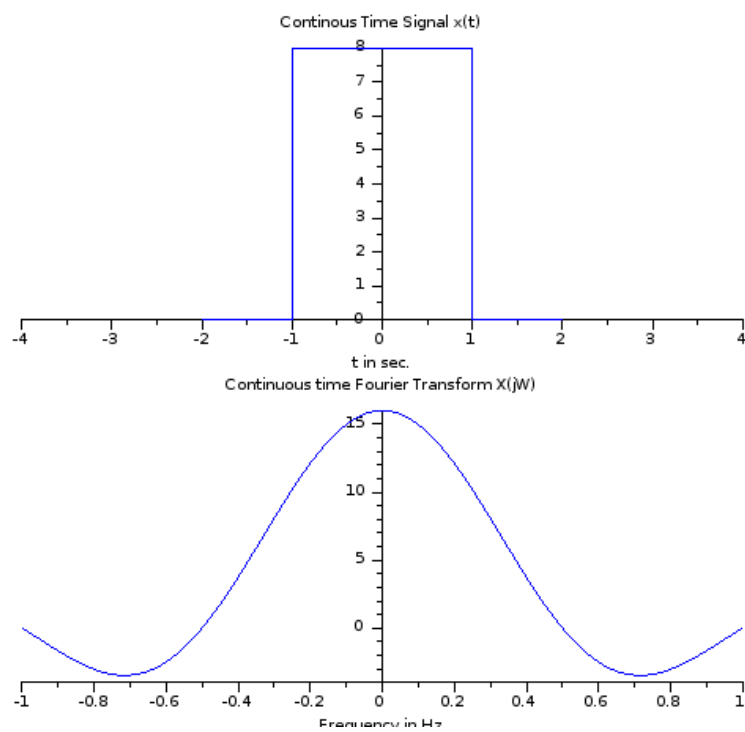


Figure 4.1: Magnitude and Phase Spectra

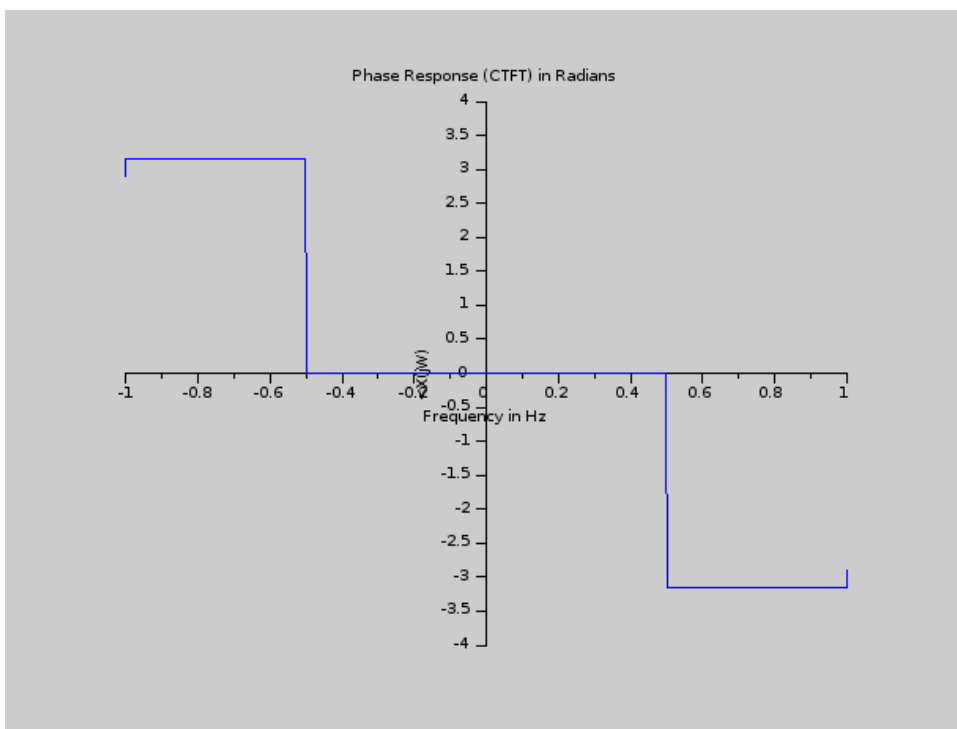


Figure 4.2: Magnitude and Phase Spectra

```

15         end
16     end
17     Wmax = 2*%pi*1;           //Analog Frequency = 1Hz
18     K = 4;
19     k = 0:(K/1000):K;
20     W = k*Wmax/K;
21     xt = xt';
22     XW = xt* exp(-sqrt(-1)*t'*W) * Dt;
23     XW_Mag = real(XW);
24     W = [-mtlbfliplr(W), W(2:1001)]; // Omega from -
        Wmax to Wmax
25     XW_Mag = [mtlbfliplr(XW_Mag), XW_Mag(2:1001)];
26     [ XW_Phase ,db] = phasemag (XW);
27     XW_Phase =[-mtlbfliplr(XW_Phase),XW_Phase(2:1001)];
28     // Plotting the Function
29     subplot(2,1,1);
30     a=gca();
31     a.data_bounds=[-4,0;4,2];
32     a.y_location="origin";
33     plot(t,xt);
34     xlabel('t in sec. ');
35     title('Continous Time Signal x(t)');
36     // Plotting Magnitude Reponse of CTS
37     subplot(2,1,2);
38     a=gca();
39     a.y_location="origin";
40     plot(W/(2*%pi),XW_Mag);
41     xlabel('Frequency in Hz');
42     title('Continuous time Fourier Transform X(jW)');
43     // Plotting Phase Reponse of CTS
44     figure(1);
45     a = gca ();
46     a.y_location = "origin";
47     a.x_location = "origin";
48     plot (W/(2*%pi),-XW_Phase *%pi /180) ;
49     xlabel ( ' Frequency in Hz' );
50     ylabel ( '<X(jW) ' )
51     title ( ' Phase Response (CTFT) in Radians ' )

```

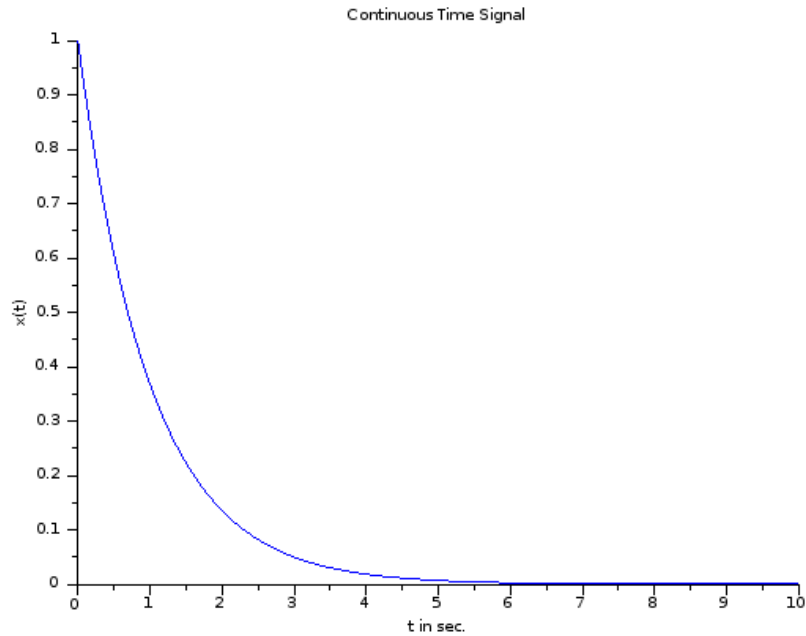


Figure 4.3: Magnitude and Phase Spectra

Scilab code Exa 4.2 Magnitude and Phase Spectra

```

1 //Scilab Code for Example 4.2 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Continuous Time Fourier Transform of a
4 //Continuous Time Signal  $x(t) = \exp(-A*t)u(t)$ ,  $t > 0$ 
5 clear;
```

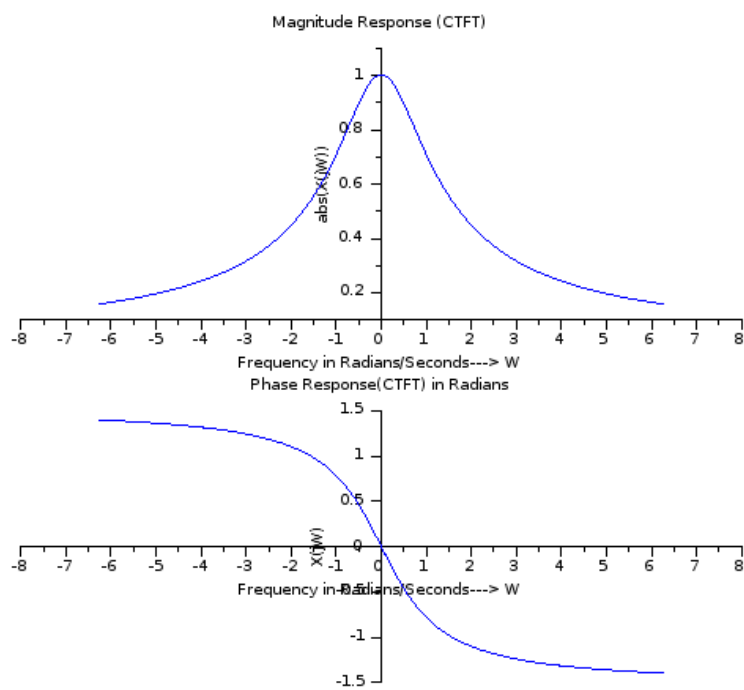


Figure 4.4: Magnitude and Phase Spectra

```

6  clc;
7  close;
8  // Analog Signal
9  A =1;    //Amplitude
10 T=1;
11 Dt = 0.005;
12 t = 0:Dt:10;
13 xt = A*exp(-t/T);
14 // Continuous-time Fourier Transform
15 Wmax = 2*%pi*1;    //Analog Frequency = 1Hz
16 K = 4;
17 k = 0:(K/1000):K;
18 W = k*Wmax/K;
19 XW = xt* exp(-sqrt(-1)*t'*W) * Dt;
20 XW_Mag = abs(XW);
21 W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
22 XW_Mag = [mtlb_fliplr(XW_Mag), XW_Mag(2:1001)];
23 [XW_Phase,db] = phasemag(XW);
24 XW_Phase = [-mtlb_fliplr(XW_Phase),XW_Phase(2:1001)
    ];
25 //Plotting Continuous Time Signal
26 a = gca();
27 a.y_location = "origin";
28 plot(t,xt);
29 xlabel('t in sec. ');
30 ylabel('x(t)')
31 title('Continuous Time Signal')
32 figure
33 //Plotting Magnitude Response of CTS
34 subplot(2,1,1);
35 a = gca();
36 a.y_location = "origin";
37 plot(W,XW_Mag);
38 xlabel('Frequency in Radians/Seconds—> W');
39 ylabel('abs(X(jW))')
40 title('Magnitude Response (CTFT)')
41 //Plotting Phase Reponse of CTS

```

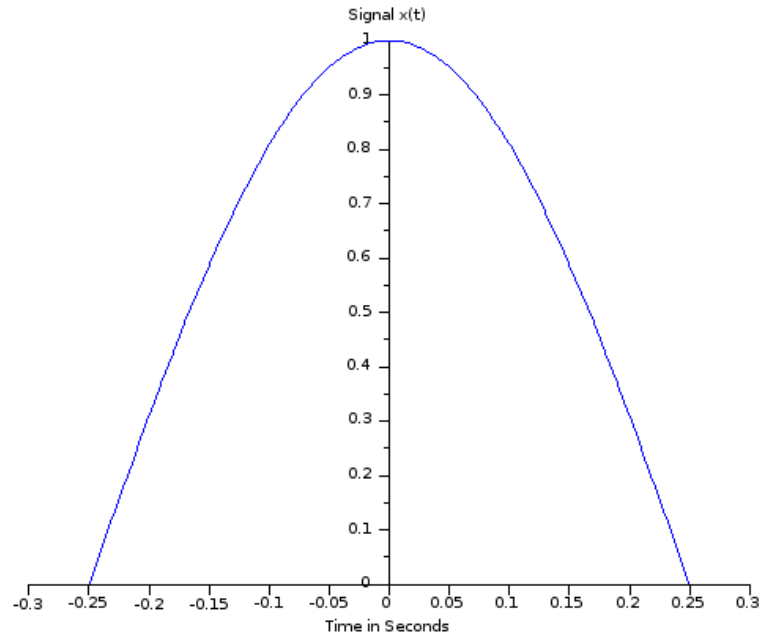


Figure 4.5: Magnitude and Phase Spectra

```

42 subplot(2,1,2);
43 a = gca();
44 a.y_location = "origin";
45 a.x_location = "origin";
46 plot(W,XW_Phase*%pi/180);
47 xlabel('                                Frequency in
         Radians/Seconds—> W');
48 ylabel('
         (jW)')
49 title('Phase Response(CTFT) in Radians')

```

<X

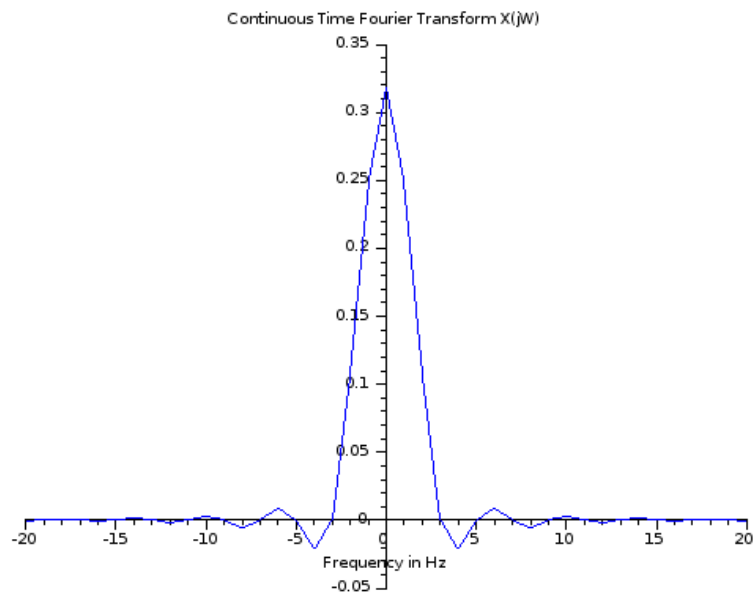


Figure 4.6: Magnitude and Phase Spectra

Scilab code Exa 4.3 Magnitude and Phase Spectra

```
1 //Scilab Code for Example 4.3 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //x(t)=A*cos w0*t*gate(t/T)
4 //T=1/2*fo
5 //fo=0.5 Hz
6 clear;
7 clc;
8 //Fourier Transform
9 A=1;
10 T=0.5;
11 fo=1/(2*T);
12 Wo=2*pi*fo;
13 for f=-20:1:20;
14 X(f+21)=A*integrate('cos(Wo*t)*cos(2*pi*f*t)', 't'
    , -0.25, 0.25);
15 end
16 disp(X, 'X(0)--->X(20)');
17 t=-0.25:0.01:0.25;
18 q=cos(Wo*t);
19 a = gca ();
20 a.y_location = "origin";
21 a.x_location = "origin";
22 plot(t,q);
23 xlabel('Time in Seconds');
24 title('Signal x(t)');
25 figure(1);
26 a = gca ();
27 a.y_location = "origin";
28 a.x_location = "origin";
29 f=-20:1:20;
30 plot(f, X);
```



```
31 xlabel ( 'Frequency in Hz ' );
32 title ( 'Continuous Time Fourier Transform X(jW) ' );
```

Scilab code Exa 4.4 Energy at output

```
1 //Scilab Code for Example 4.4 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 //X(f)=A*T/(1+j*2*pi*f*T)
6 syms f w;
7 A=1;
8 T=1;
9 X=(A^2*T^2)/(1+4*pi^2*f^2*T^2)
10 disp('Putting f = tan @');
11 disp('Total Energy:');
12 Ex=integrate(' (A^2*T)/(2*pi)', 'w', -pi/2, pi/2)
13 disp('Energy Contained in the Output Signal');
14 Ey=integrate(' (A^2*T)/(2*pi)', 'w', -pi/4, pi/4)
15 e=Ey*100/Ex;
16 disp(e, 'Percentage Energy Contained in the Output:');
  ;
```

Scilab code Exa 4.5 Fourier Transform

```
1 //Scilab Code for Example 4.5 of Signals and systems
  by
2 //P.Ramakrishna Rao
```

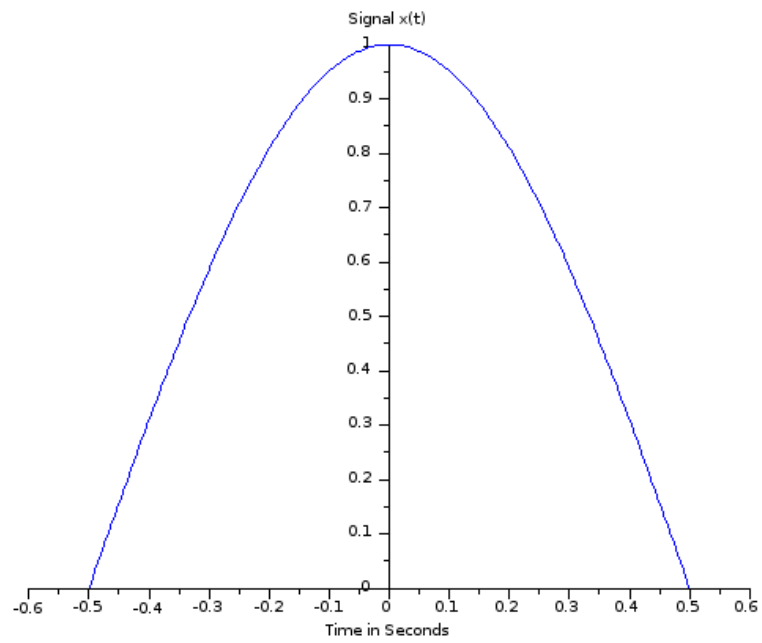


Figure 4.7: Fourier Transform

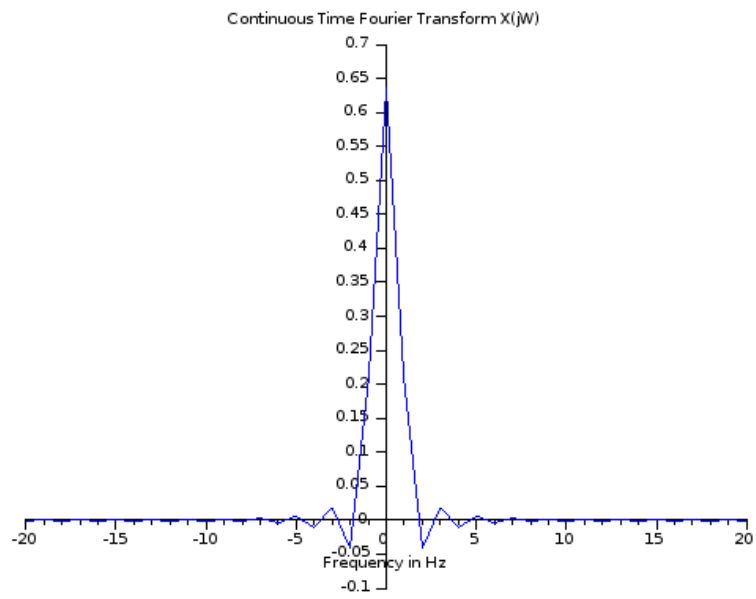


Figure 4.8: Fourier Transform

```

3 //x(t)=cos pi*t, |t|>0.5, zero otherwise
4 clear;
5 clc;
6 //Fourier Transform
7 for f=-20:1:20;
8 X(f+21)=integrate('cos(%pi*t)*cos(2*%pi*f*t)', 't'
    , -0.5, 0.5);
9 end
10 disp(X, 'X(0)-->X(20)');
11 t=-0.5:0.01:0.5;
12 q=cos(%pi*t);
13 a = gca ();
14 a.y_location = "origin";
15 a.x_location = "origin";
16 plot(t,q);
17 xlabel('Time in Seconds');
18 title('Signal x(t)');
19 figure(1);
20 a = gca ();
21 a.y_location = "origin";
22 a.x_location = "origin";
23 f=-20:1:20;
24 plot(f, X);
25 xlabel('Frequency in Hz');
26 title('Continuous Time Fourier Transform X(jW)');

```

Scilab code Exa 4.6 Convolution

```

1 //Scilab Code for Example 4.6 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clear;
4 clc;

```

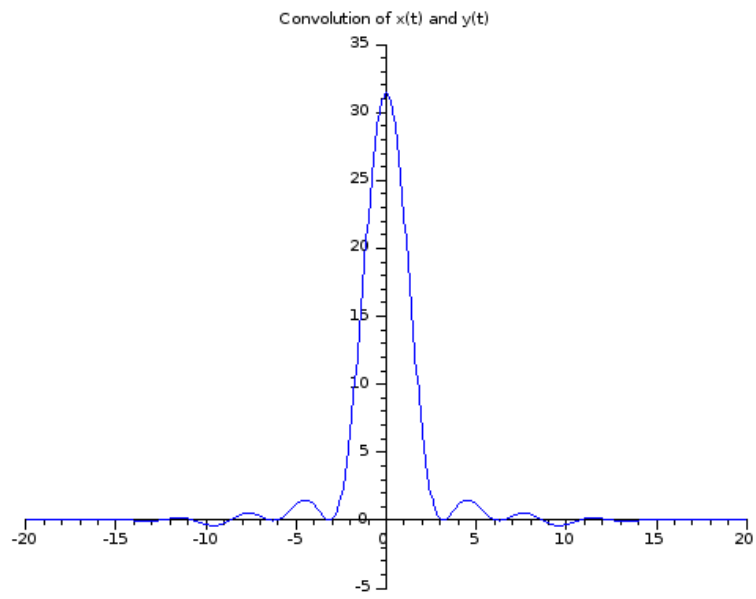


Figure 4.9: Convolution

```

5 close;
6 disp("X(w)=1/(a+jw)^2=1/(a+jw)*1/(a+jw)");
7 disp("exp(-a*t)*u(t) <--> 1/(a+jw)");
8 disp("therefore x(t)=convolution(2*sinc(2*t),(sinc t
    )^2");
9 a=2;
10 n=1;
11 for t=-10:0.1:10
12 y1(1,n)=(sinc(t))^2;
13 y2(1,n)=2*sinc(2*t);
14 n=n+1;
15 z=conv(y1,y2);
16 end
17 t=-20:0.1:20;
18 a=gca();
19 a.y_location="origin";
20 a.x_location="origin";
21 plot(t,z);
22 title('Convolution of x(t) and y(t)')

```

Scilab code Exa 4.7 Energy in Signal

```

1 //Scilab Code for Example 4.7 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms t;
6 A=2;
7 //x(t)=20*sinc(10*t);
8 disp("Total Energy of the signal:");
9 q=integrate('A^2','w',-5,5);
10 disp(q,'Ex');

```

Scilab code Exa 4.11 Fourier Transform

```
1 //Scilab Code for Example 4.11 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Find system function and output of the system
4 clear;
5 clc;
6 syms a t;
7 for n=1:10;
8 h=t^(n-1)*%e^(-a*t);
9 H=laplace(h,t,'jw');
10 disp(H,'SYSTEM FUNCTION X(f)=',h,'For x(t)=');
11 end
```

Scilab code Exa 4.14 Fourier Transform

```
1 //Scilab Code for Example 4.14 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 close;
6 a=.5;
7 A=1/(sqrt(2)*%pi);
8 t=-10:0.1:10;
9 x=A*exp(-a*t.*t);
```

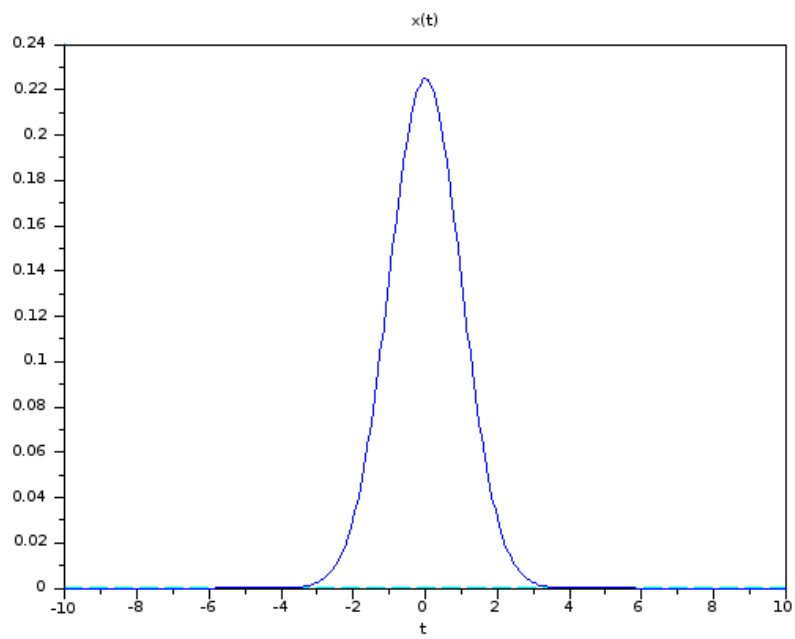


Figure 4.10: Fourier Transform

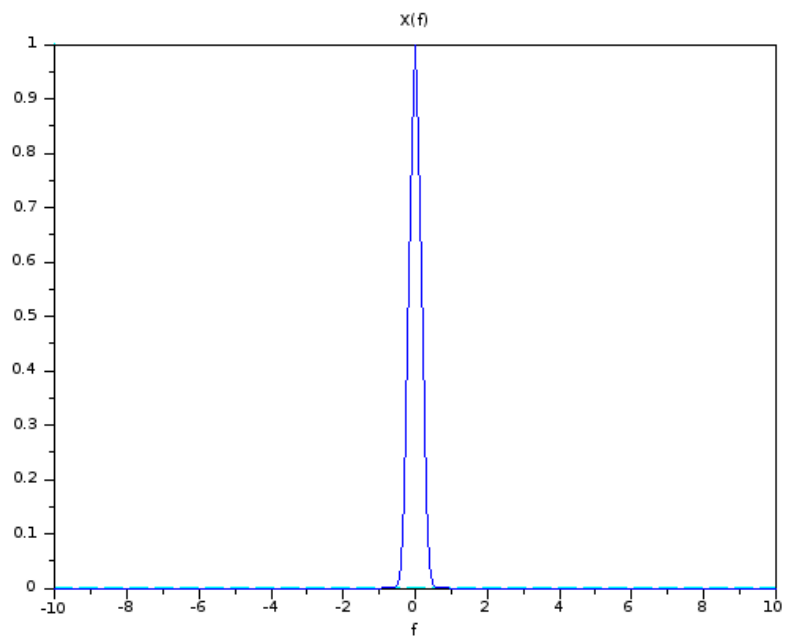


Figure 4.11: Fourier Transform

```

10 disp(" Guassian pulse signal  $x(t)=(1/\sqrt{2})*\pi*\exp$ 
      ( $-a*t^2$ )");
11 disp("X(w)=integral (exp(-a*t^2)*exp(-%i*w*t)) w.r.t
      dt");
12 disp("d(X(w))/dw=-%i*w/(2*a)*integral (exp(-a*t^2)*
      exp(-%i*w*t))");
13 disp("d(X(w))/dw=-w*X(w)/2a");
14 disp("solving this we get  $X(w)=A*\exp(-w^2/4a)$ ")
15 disp("A=sqrt(%pi/a)");
16 d=gca();
17 plot(t,x);
18 poly1=d.children.children;
19 poly1.thickness=3;
20 poly1.foreground=2;
21 xtitle('x(t)', 't')
22 A=1;
23 f=t;
24 Xf=A*exp(-2*pi^2*f^2);
25 figure(1);
26 d=gca();
27 plot(f,Xf);
28 poly1=d.children.children;
29 poly1.thickness=3;
30 poly1.foreground=2;
31 xtitle('X(f)', 'f')

```

Scilab code Exa 4.17 Fourier Transform

```

1 //Scilab Code for Example 4.17 of Signals and
      systems by
2 //P.Ramakrishna Rao

```

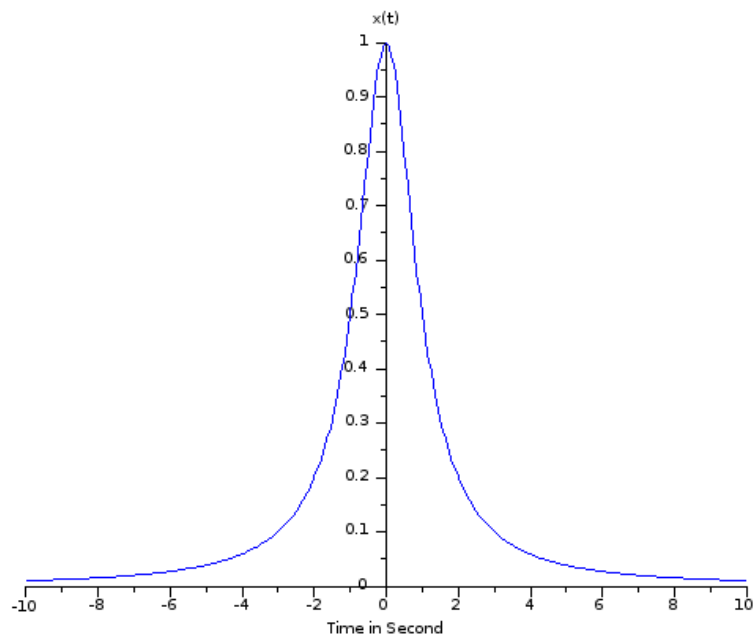


Figure 4.12: Fourier Transform

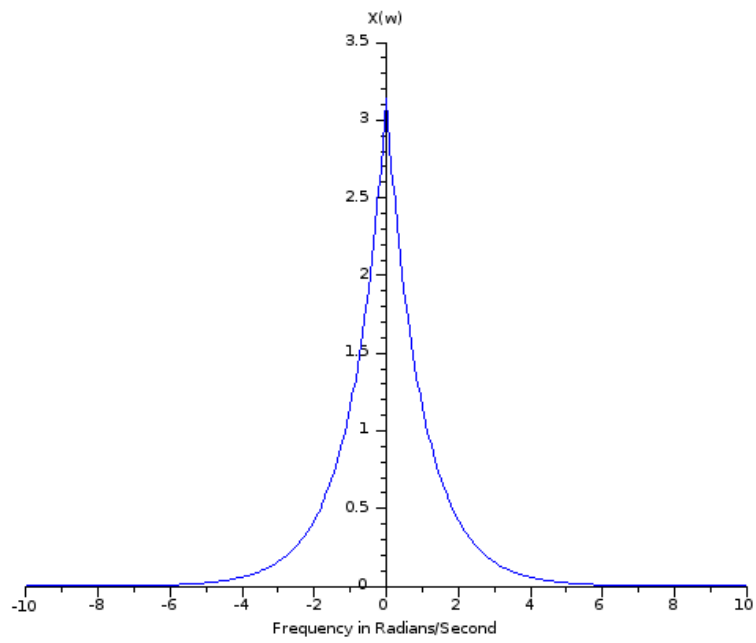


Figure 4.13: Fourier Transform

```

3 clear;
4 clc;
5 //Fourier Transform of x(t)
6 n=1;
7 for t=-10:0.1:10;
8     x(1,n)=1/(1+t^2);
9     n=n+1;
10 end
11 a = gca ();
12 a.y_location = "origin";
13 a.x_location = "origin";
14 t=-10:0.1:10;
15 plot(t,x);
16 xlabel ( 'Time in Second' );
17 title ( 'x(t)' );
18 disp("By Duality Theorem Fourier Transform of x(t)
      is:");
19 disp("%pi*exp(-|w|)");
20 n=1;
21 for w=-10:0.1:10;
22     X(1,n)=%pi*exp(-abs(w));
23     n=n+1;
24 end
25 figure(1);
26 a = gca ();
27 a.y_location = "origin";
28 a.x_location = "origin";
29 w=-10:0.1:10;
30 plot(w,X);
31 xlabel ( 'Frequency in Radians/Second' );
32 title ( 'X(w)' );

```

Chapter 5

Fourier Representation of Discrete Time Signals

Scilab code Exa 5.1 DTFT

```
1 // Scilab Code for Example 5.1 of Signals and systems
  by
2 // P. Ramakrishna Rao
3 // Discrete Time Fourier Transform of
4 //  $x[n] = (a^{abs(n)})$   $0 < a < 1$ 
5 clear;
6 clc;
7 close;
8 // DTS Signal
9 a = 0.5; //  $0 < a < 1$ 
10 max_limit = 10;
11 n = -max_limit+1:max_limit-1;
12 x = a^abs(n);
13 // Discrete-time Fourier Transform
14 Wmax = 2*%pi;
15 K = 4;
16 k = 0:(K/1000):K;
17 W = k*Wmax/K;
18 XW = x* exp(-sqrt(-1)*n'*W);
```

```

19 XW_Mag = real(XW);
20 W = [-mtlbfliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
21 XW_Mag = [mtlbfliplr(XW_Mag), XW_Mag(2:1001)];
22 a = gca();
23 a.y_location = "origin";
24 a.x_location = "origin";
25 plot2d3('gnn',n,x);
26 xtitle('Discrete Time Sequence x[n] for a>0')
27 figure(1);
28 a = gca();
29 a.y_location = "origin";
30 a.x_location = "origin";
31 plot2d(W,XW_Mag);
32 title('Discrete Time Fourier Transform X(exp(jW))')

```

Scilab code Exa 5.2 DTFT and Spectra

```

1 //Scilab Code for Example 5.2 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Discrete Time Fourier Transform of
4 //x[n]= 1 , 0=<n<=3
5 clear;
6 clc;
7 close;
8 // DTS Signal
9 N1 = 3;
10 n = 0:N1;
11 x = ones(1,length(n));
12 // Discrete-time Fourier Transform

```

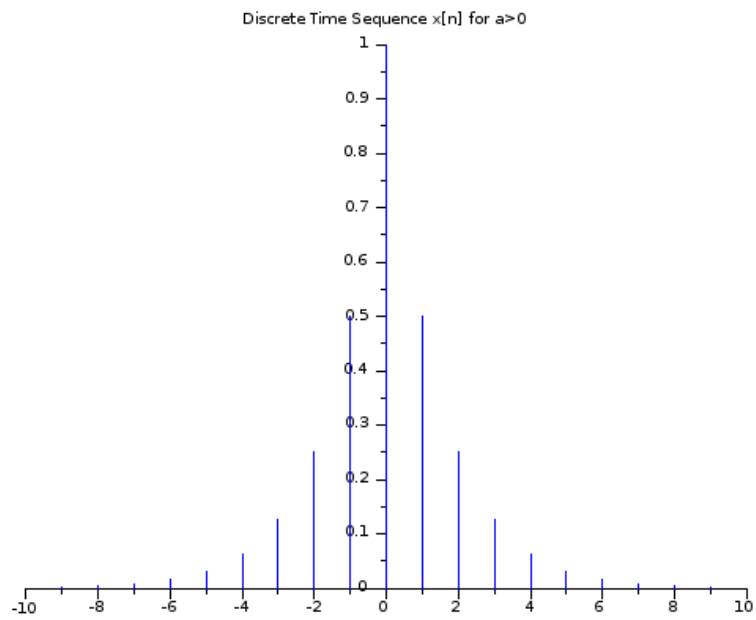


Figure 5.1: DTFT

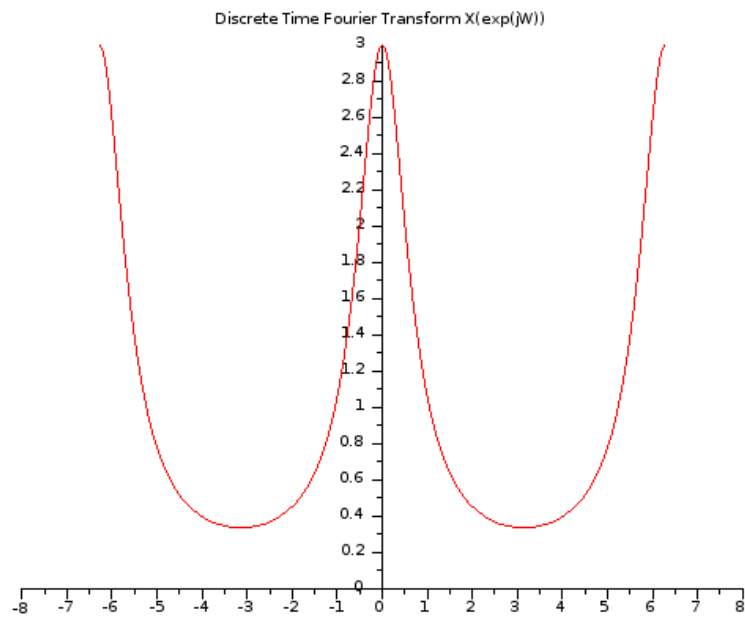


Figure 5.2: DTFT

```

13 Wmax = 2*%pi;
14 K = 4;
15 k = 0:(K/1000):K;
16 W = k*Wmax/K;
17 XW = x* exp(-sqrt(-1)*n'*W);
18 XW_Mag = real(XW);
19 [XW_Phase,db] = phasemag(XW);
20 W = [-mtlbfliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
21 XW_Mag = [mtlbfliplr(XW_Mag), XW_Mag(2:1001)];
22 XW_Phase = [-mtlbfliplr(XW_Phase),XW_Phase(2:1001)
    ];
23 a = gca();
24 a.y_location = "origin";
25 a.x_location = "origin";
26 plot2d3('gnn',n,x);
27 xtitle('Discrete Time Sequence x[n]')
28 figure(1);
29 a = gca();
30 a.y_location = "origin";
31 a.x_location = "origin";
32 plot2d(W,abs(XW_Mag));
33 title('Discrete Time Fourier Transform X(exp(jW))')
34 figure(2);
35 a = gca();
36 a.y_location = "origin";
37 a.x_location = "origin";
38 plot2d(W,XW_Phase);
39 title('Phase Response <(X(jW))')

```

Scilab code Exa 5.3 Inverse Fourier

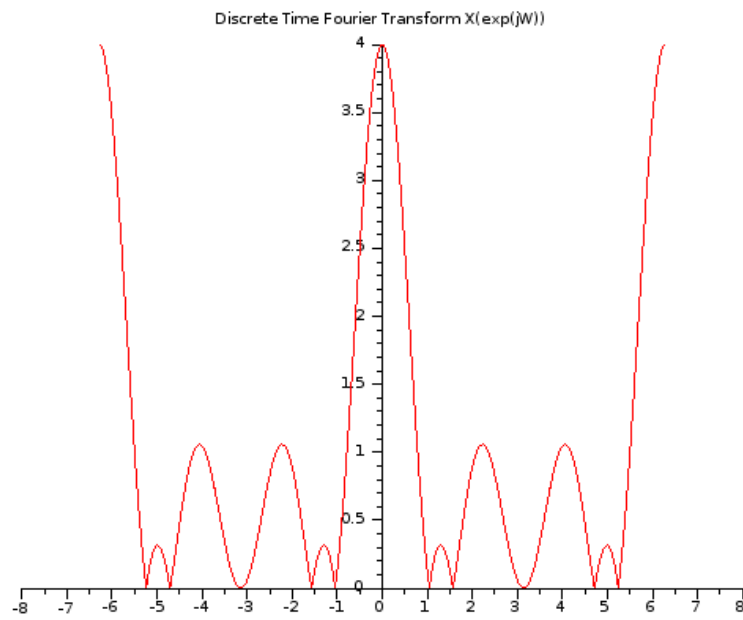


Figure 5.3: DTFT and Spectra

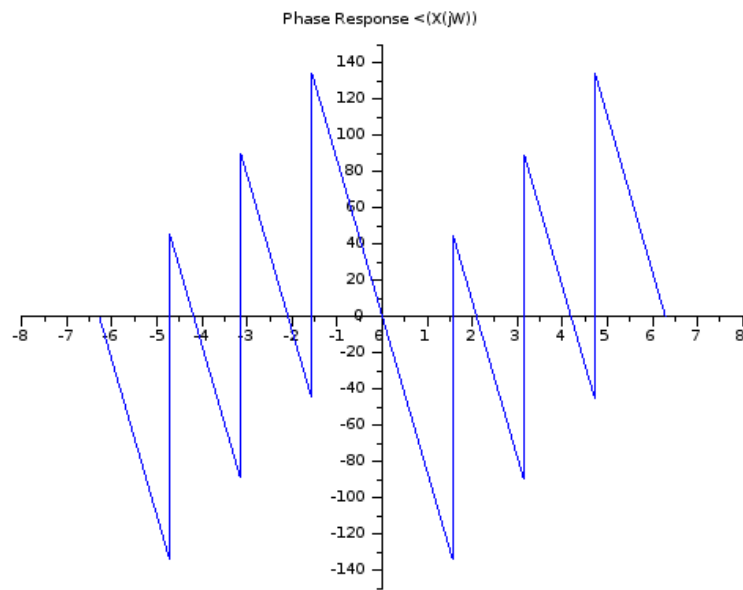


Figure 5.4: DTFT and Spectra

```

1 //Scilab Code for Example 5.3 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 //Inverse Fourier Transform
6 w=-20:20;
7 X=4*(cos(w))^2;
8 a = gca();
9 a.y_location = "origin";
10 a.x_location = "origin";
11 plot(w,X);
12 title("X(e^jw)");
13 xlabel("Frequency in Radians/sec");
14 figure(1);
15 n=-50:50;
16 x=2*sinc(n)+sinc(n+2)+sinc(n-2);
17 a = gca ();
18 a.y_location = "origin";
19 a.x_location = "origin";
20 plot2d3(n,x);
21 title("x(n)");
22 xlabel("Time in sec");

```

Scilab code Exa 5.4 Inverse Fourier

```

1 //Scilab Code for Example 5.4 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Discrete Time Fourier Transform of
4 //  $X(e^{j\omega})=2\pi\delta(\omega)$ 

```

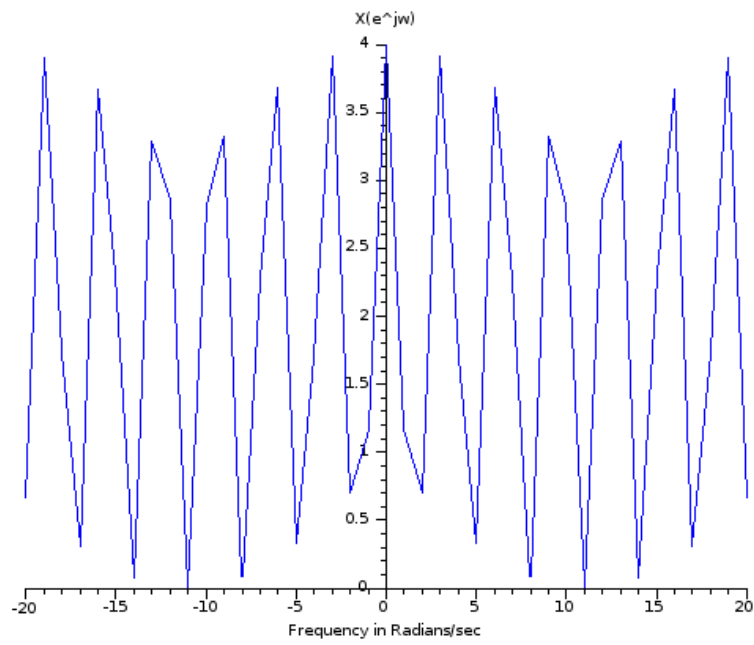


Figure 5.5: Inverse Fourier

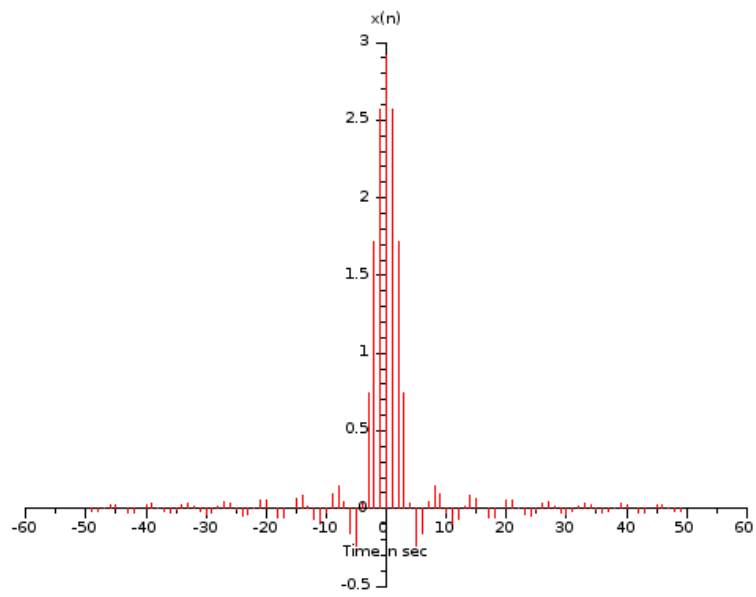


Figure 5.6: Inverse Fourier

```

5 clear;
6 clc;
7 close;
8 N = 1;
9 N1 = -3*N:3*N;
10 xn = [zeros(1,N-1),1];
11 x = [1 xn xn xn xn xn xn];
12 ak = 1/N;
13 XW = 2*%pi*ak*ones(1,2*N);
14 Wo = 2*%pi/N;
15 n = -N:N-1;
16 W = Wo*n;
17 a = gca();
18 a.y_location = "middle";
19 a.x_location = "origin";
20 plot2d3('ggn',W,XW,2);
21 poly1 = a.children(1).children(1);
22 poly1.thickness = 3;
23 xlabel('W');
24 title('DTFT of Periodic Impulse Train')
25 figure(1);
26 a = gca();
27 a.y_location = "origin";
28 a.x_location = "origin";
29 plot2d3('ggn',N1,x,2);
30 poly1 = a.children(1).children(1);
31 poly1.thickness = 3;
32 xlabel('n');
33 title('Periodic Impulse Train x(n)')

```

Scilab code Exa 5.5 Fourier series

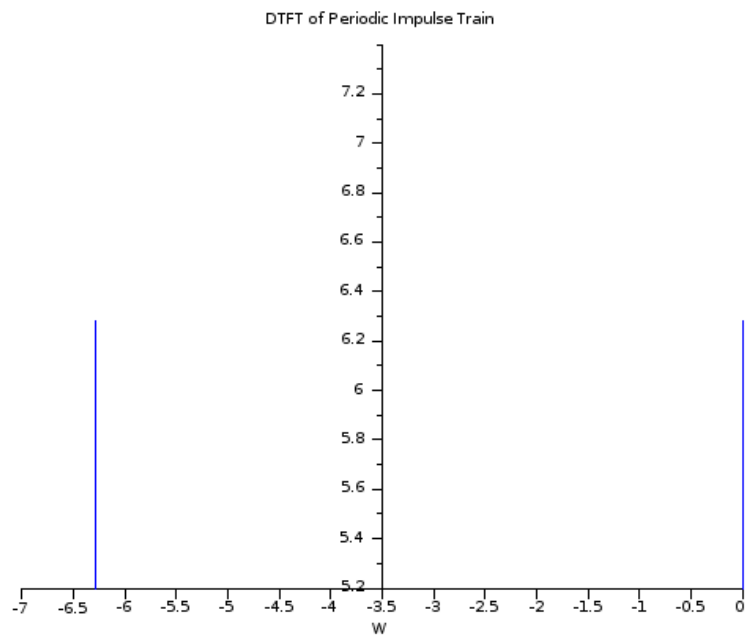


Figure 5.7: Inverse Fourier

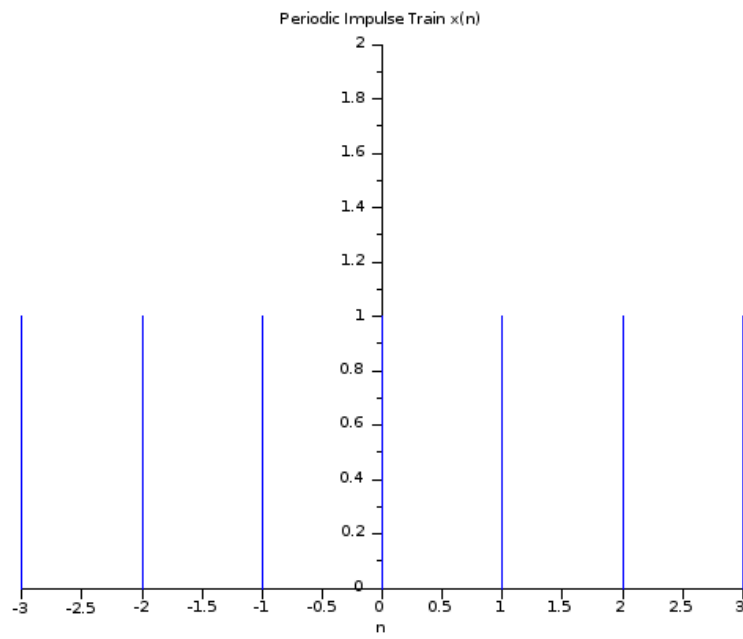


Figure 5.8: Inverse Fourier

```

1 //Scilab Code for Example 5.5 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 x=[1,2,3,2,1];
6 q=0;
7 for n=1:5
8     q=x(n)+q;
9 end
10 disp(q, '(a) X(e^j*0)');
11 q=0;
12 for n=-2:2
13     q=(-1)^n*x(n+3)+q;
14 end
15 disp(q, '(c) X(e^j*pi)');
16 disp('(d) X(e^j*pi)=2*pi*x(0)');
17 disp(2*pi*x(3));
18 q=0;
19 for n=-2:2
20     q=(x(n+3))^2+q;
21 end
22 disp(q*2*pi, '(e) |X(e^j*w)|^2');

```

Scilab code Exa 5.9.i DTFT

```

1 //Scilab Code for Example 5.9(i) of Signals and
  systems by
2 //P.Ramakrishna Rao
3 clc;
4 clear;
5 q=0;
6 a1=0.5;
7 a2=-0.5;
8 n=1:101;

```

```

9  x1=a1^n;
10 x2=a2^n;
11 w=2;
12 n=0:100;
13 z=(exp(-%i*w*n));
14 for n=0:33;
15     X(n+1)=z(n+1)*x1(3*n+1);
16     q=X(n+1)+q;
17 end
18 disp(q, 'Y1(e^2j) at a=0.5 ');
19 for n=0:33;
20     X(n+1)=z(n+1)*x2(3*n+1);
21     q=X(n+1)+q;
22 end
23 disp(q, 'Y1(e^2j) at a=-0.5 ');

```

Scilab code Exa 5.9.ii DTFT

```

1 //Scilab Code for Example 5.9(ii) of Signals and
  systems by
2 //P.Ramakrishna Rao
3 clc;
4 clear;
5 q=0;
6 a1=0.5
7 a2=-0.5;
8 n=1:101;
9 x1=a1^n;
10 x2=a2^n;
11 w=2;
12 n=0:100;
13 z=(exp(-%i*w*n));
14 for n=0:33;
15     X(n+1)=z(n+1)*x1(n+1);
16     q=X(n+1)+q;

```

```

17 end
18 disp(q, 'Y2(e^2j) at a=0.5 ');
19 for n=0:33;
20     X(n+1)=z(n+1)*x2(n+1);
21     q=X(n+1)+q;
22 end
23 disp(q, 'Y2(e^2j) at a=-0.5 ');

```

Scilab code Exa 5.9.iii DTFT

```

1 //Scilab Code for Example 5.9(iii) of Signals and
   systems by
2 //P.Ramakrishna Rao
3 clc;
4 clear;
5 q=0;
6 a1=0.5
7 a2=-0.5;
8 n=1:101;
9 x1=a1^n;
10 x2=a2^n;
11 w=2;
12 n=0:100;
13 z=(exp(-%i*w*n));
14 for n=0:33;
15     X(n+1)=z(n+1)*x1(n+1)*cos(0.4*%pi*n);
16     q=X(n+1)+q;
17 end
18 disp(q, 'Y3(e^2j) at a=0.5 ');
19 for n=0:33;
20     X(n+1)=z(n+1)*x2(n+1)*cos(0.4*%pi*n);
21     q=X(n+1)+q;
22 end
23 disp(q, 'Y3(e^2j) at a=-0.5 ');

```

Scilab code Exa 5.11 Inverse Fourier

```
1 //Scilab Code for Example 5.11 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 wc=1;
6 y=1;
7 for n=-%pi:%pi/80:%pi
8     if n<-wc | n>wc then
9         X(1,y)=1;
10        y=y+1;
11    else X(1,y)=0;
12        y=y+1;
13    end
14 end
15 n=-%pi:%pi/80:%pi;
16 a = gca ();
17 a.y_location = "origin";
18 a.x_location = "origin";
19 plot(n,X);
20 xlabel ( 'Frequency in Radians/Seconds' );
21 title ( 'X(e^jw) at Wc=1' );
22 A=1/%pi;
23 for k=-10:10
24     x(k+11)=A*integrate( 'cos(w*k)', 'w', wc, %pi);
25 end
26 figure(1);
27 k=-10:10;
28 a = gca ();
29 a.y_location = "origin";
30 a.x_location = "origin";
31 plot2d3(k,x);
```

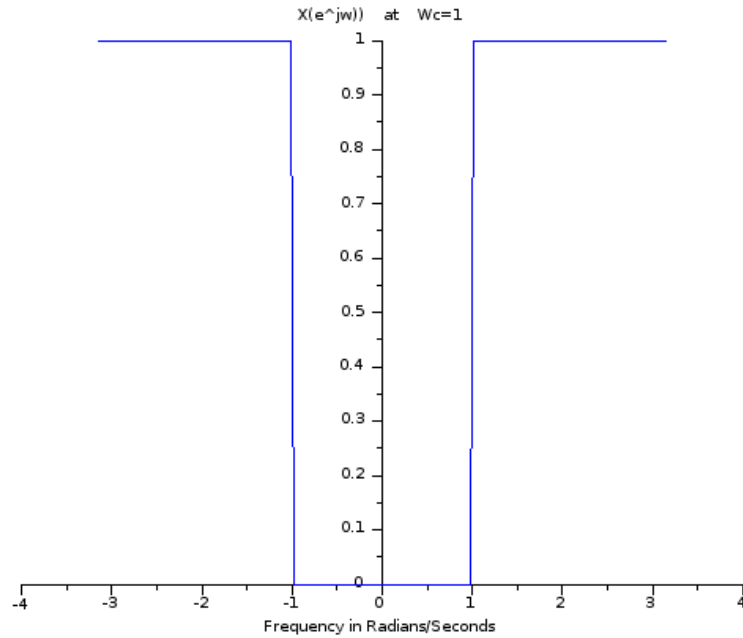


Figure 5.9: Inverse Fourier

```
32 xlabel ( 'Time in Seconds' );
33 title ( 'x(n)      at      Wc=1' );
```

Scilab code Exa 5.13 Circular Convolution

```
1 //Scilab Code for Example 5.13 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //Circular Convolution
```

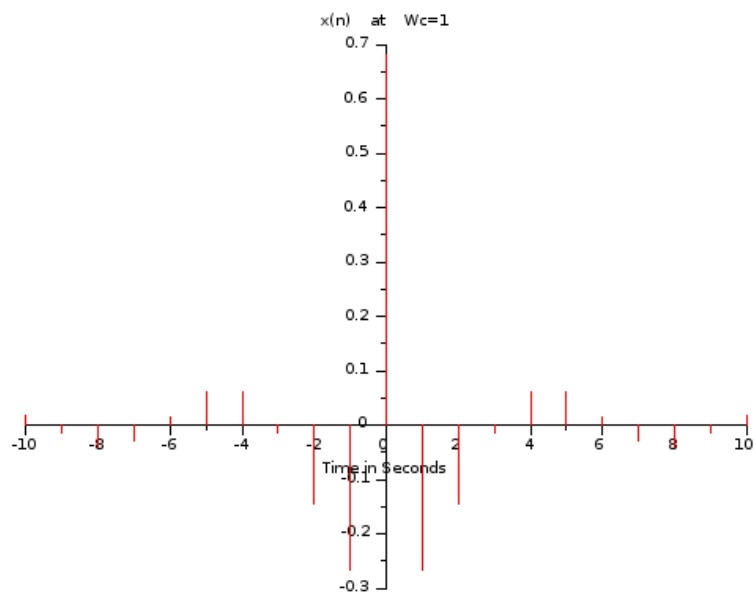


Figure 5.10: Inverse Fourier


```

4 clear;
5 //First Causal sequence
6 x=[1,4,3,2;2,1,4,3;3,2,1,4;4,3,2,1]
7 //Second Sequence
8 y=[4;3;2;1]
9 //Conolution
10 z=x*y;
11 disp(z, 'Convolution Of x & y is:')

```

Scilab code Exa 5.14 Circular Convolution

```

1 //Scilab Code for Example 5.14 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //Circular Convolution
4 clear;
5 clc;
6 close;
7 //First Causal sequence
8 x=[1,1,1,1;1,-%i,-1,%i;1,-1,1,-1;1,%i,-1,-%i]
9 y1=[1;2;3;4];
10 y2=[4;3;2;1];
11 X=x*y1;
12 disp(X, 'Vector X(0)--->X(3) ')
13 Y=x*y2;
14 disp(Y, 'Vector Y(0)--->Y(3) ')
15 for n=1:4;
16     Z(n,1)=X(n,1)*Y(n,1);
17 end
18 q=4*x^-1;
19 disp(q, 'IDFT matrix ');
20 z=0.25*q*Z;
21 disp(z, 'IDFT of Vector Z(0)--->Z(3) ')

```

Scilab code Exa 5.16 DTFT

```
1 //Scilab Code for Example 5.16 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 clc;
4 clear;
5 q=0;
6 x=[1,2,3,4];
7 w=0;
8 n=0:3;
9 z=(exp(-%i*w*n));
10 for n=0:3
11     X(n+1)=z(n+1)*x(n+1);
12     q=X(n+1)+q;
13 end
14 disp(q, 'X(0)->zeroth DFT coefficient ');
15 q=0;
16 w=%pi/2;
17 n=0:3;
18 z=(exp(-%i*w*n));
19 for n=0:3
20     X(n+1)=z(n+1)*x(n+1);
21     q=X(n+1)+q;
22 end
23 disp(q, 'X(1)->first DFT coefficient ');
24 q=0;
25 w=%pi;
26 n=0:3;
27 z=(exp(-%i*w*n));
28 for n=0:3
29     X(n+1)=z(n+1)*x(n+1);
30     q=X(n+1)+q;
31 end
```

```

32 disp(ceil(q), 'X(2)->second DFT coefficient ');
33 q=0;
34 w=3*%pi/2;
35 n=0:3;
36 z=(exp(-%i*w*n));
37 for n=0:3
38     X(n+1)=z(n+1)*x(n+1);
39     q=X(n+1)+q;
40 end
41 disp(q, 'X(3)->third DFT coefficient ');

```

Scilab code Exa 5.18 DFT coefficients

```

1 //Scilab Code for Example 5.18 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //Given signal x(n)
4 clear;
5 clc;
6 x=[1,2,3,4];
7 X=fft(x);
8 disp(X, 'FFT of given signal is :X(0)-->X(3) ');

```

Chapter 6

Sampling and Reconstruction of Bandlimited Signals

Scilab code Exa 6.1 Sampling and Reconstruction

```
1 //Scilab Code for Example 6.1 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Sampling of signal
4 clc;
5 clear f n X X_delta w;
6 fs=200;
7 for f=-200:200
8     X(f+201)=5*[delta(f-75)+delta(f+75)];
9 end
10 figure(1);
11 f=-200:200;
12 plot2d3(f,X,-2);
13 title('X(f)');
14 xlabel('----> f');
15 w=1;
16 n=-1;
17     for f=-275:275
18         X_delta1(f+276)=fs*5*[delta(f-n*fs-75)+delta(f-n
```

```

        *fs+75)];
19     end
20     n=n+1;
21     for f=-275:275
22         X_delta2(f+276)=fs*5*[delta(f-n*fs-75)+delta(f-n
            *fs+75)];
23     end
24     n=n+1;
25     for f=-275:275
26         X_delta3(f+276)=fs*5*[delta(f-n*fs-75)+delta(f-n
            *fs+75)];
27     end
28     X_delta=X_delta1+X_delta2+X_delta3;
29     figure(2);
30     f=-275:275;
31     plot2d3(f,X_delta,-2);
32     title('X_delta(f) at fs=200');
33     xlabel('----> f');

```

Scilab code Exa 6.2 Sampling and Reconstruction

```

1 //Scilab Code for Example 6.2 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Sampling of signal and aliasing due to low
  Sampling frequency
4 clc;
5 clear f n X X_delta X_delta1 X_delta2 X_delta3 w;
6 fs=100;
7 for f=-200:200
8     X(f+201)=5*[delta(f-75)+delta(f+75)];

```

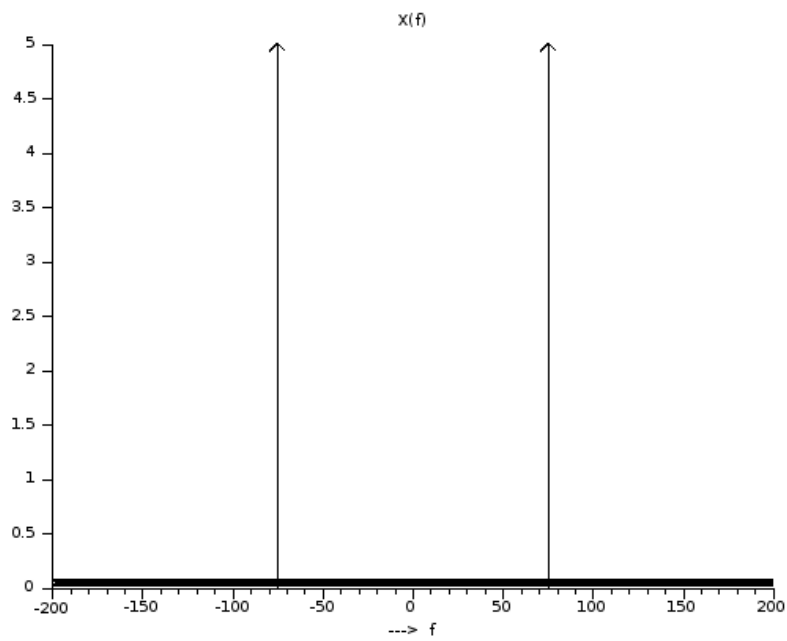


Figure 6.1: Sampling and Reconstruction

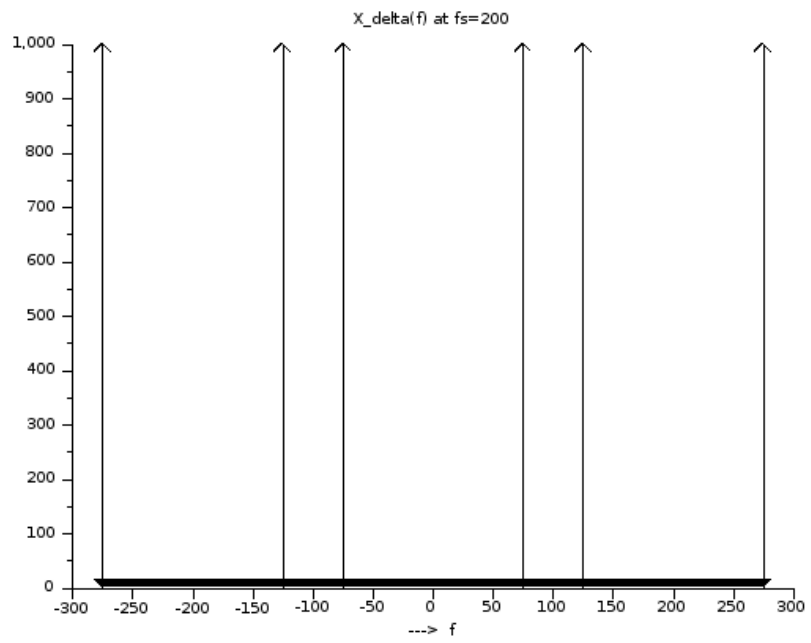


Figure 6.2: Sampling and Reconstruction

```

9  end
10 figure(1);
11 f=-200:200;
12 plot2d3(f,X,-2);
13 title('X(f)');
14 xlabel('----> f');
15 w=1;
16 n=-2;
17     for f=-175:175
18         X_delta1(f+176)=fs*5*[delta(f-n*fs-75)+delta(f-n
19             *fs+75)];
20     end
21     n=n+1;
22     for f=-175:175
23         X_delta2(f+176)=fs*5*[delta(f-n*fs-75)+delta(f-n
24             *fs+75)];
25     end
26     n=n+1;
27     for f=-175:175
28         X_delta3(f+176)=fs*5*[delta(f-n*fs-75)+delta(f-n
29             *fs+75)];
30     end
31     n=n+1;
32     for f=-175:175
33         X_delta4(f+176)=fs*5*[delta(f-n*fs-75)+delta(f-n
34             *fs+75)];
35     end
36     n=n+1;
37     for f=-175:175
38         X_delta5(f+176)=fs*5*[delta(f-n*fs-75)+delta(f-n
39             *fs+75)];
40     end
41     X_delta=X_delta1+X_delta2+X_delta3+X_delta4+X_delta5
42         ;
43     figure(2);
44     f=-175:175;
45     plot2d3(f,X_delta,-2);

```

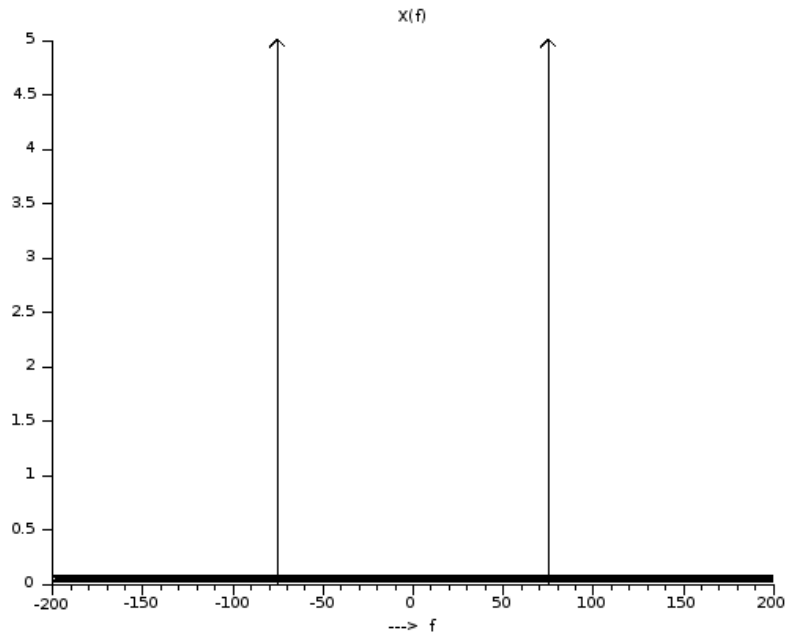



Figure 6.3: Sampling and Reconstruction

```

41 title('X_delta(f) at fs=100');
42 xlabel('----> f');
43 //The Presence of the 25 HZ component in the
    spectrum of x_delta(t)
44 //Due to effect of Aliasing

```

Scilab code Exa 6.3 Minimum Number of samples

```

1 //Scilab Code for Example 6.3 of Signals and systems

```

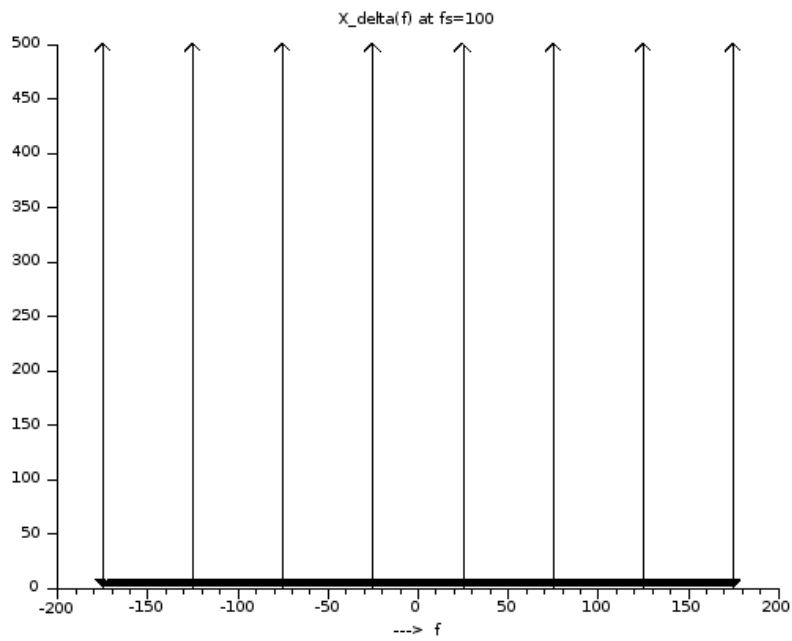


Figure 6.4: Sampling and Reconstruction

```

    by
2 //P.Ramakrishna Rao
3 //Minimum no. of samples required
4 clc;
5 clear;
6 syms t;
7 x=10*cos(6*%pi*t)+4*sin(8*%pi*t);
8 disp(x, 'x(t)');
9 disp('Maximum Frequency component present: 4 Hz');
10 fs=2*4;
11 disp(fs, 'Minimum Sampling Frequency to avoid
    distortion is:');

```

Scilab code Exa 6.4 Minimum Sampling Frequency

```

1 //Scilab Code for Example 6.4 of Signals and systems
    by
2 //P.Ramakrishna Rao
3 //Sampling Frequency
4 clc;
5 clear;
6 //x(t)=100*(sinc(100*t))^2;
7 fo=100;
8 fs=2*fo;
9 disp(fs, 'Minimum Sampling Frequency to avoid
    distortion is:');

```

Scilab code Exa 6.5 LPF output

```

1 //Scilab Code for Example 6.5 of Signals and systems
    by
2 //P.Ramakrishna Rao//Output of LPF
3 clc;

```

```

4 clear;
5
6 for f=-100:100
7     X(f+101)=delta(f+100)+delta(f-100)+3*[delta(f
          +90)+delta(f-90)];
8 end
9 f=-100:100;
10 a=gca();
11 a.x_location="origin";
12 a.y_location="origin";
13 plot2d3(f,X,-2);
14 title('Spectrum of x(t)');
15 xlabel('----> f');
16 fs=150;
17 n=-1;f=n*fs
18     for f=-275:275
19         X_delta1(f+276)=delta(f-n*fs+100)+delta(f-n*fs
          -100)+3*[delta(f-n*fs+90)+delta(f-n*fs-90)];
20     end
21 n=n+1;
22     for f=-275:275
23         X_delta2(f+276)=delta(f-n*fs+100)+delta(f-n*fs
          -100)+3*[delta(f-n*fs+90)+delta(f-n*fs-90)];
24 end
25 n=n+1;
26     for f=-275:275
27         X_delta3(f+276)=delta(f-n*fs+100)+delta(f-n*fs
          -100)+3*[delta(f-n*fs+90)+delta(f-n*fs-90)];
28 end
29 X_delta=X_delta1+X_delta2+X_delta3;
30 figure(2);
31 f=-275:275;
32 plot2d3(f,X_delta,-2);
33 title('X_delta(f) at fs=150');
34 xlabel('----> f');

```

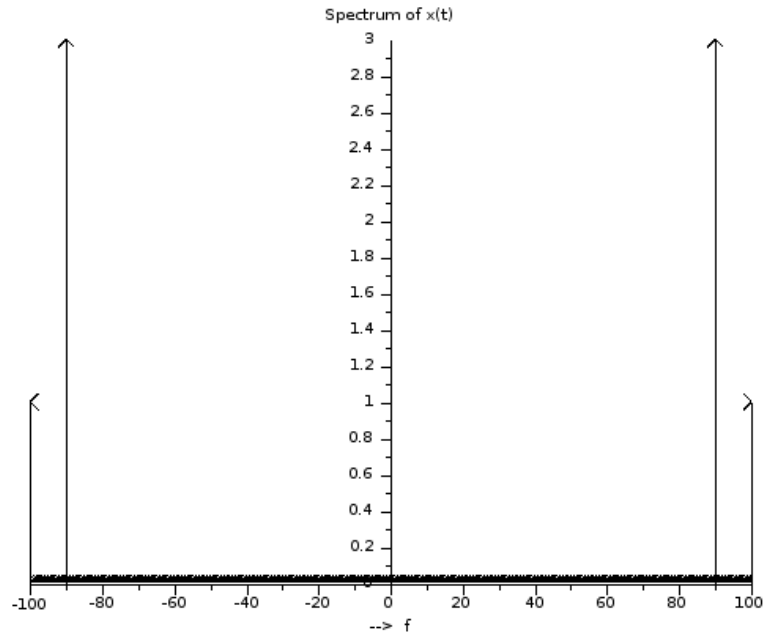


Figure 6.5: LPF output

Scilab code Exa 6.6 Sampling Frequency

```

1 //Scilab Code for Example 6.6 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Sampling Frequency / Nyquist Rate
4 clc;
5 clear;

```

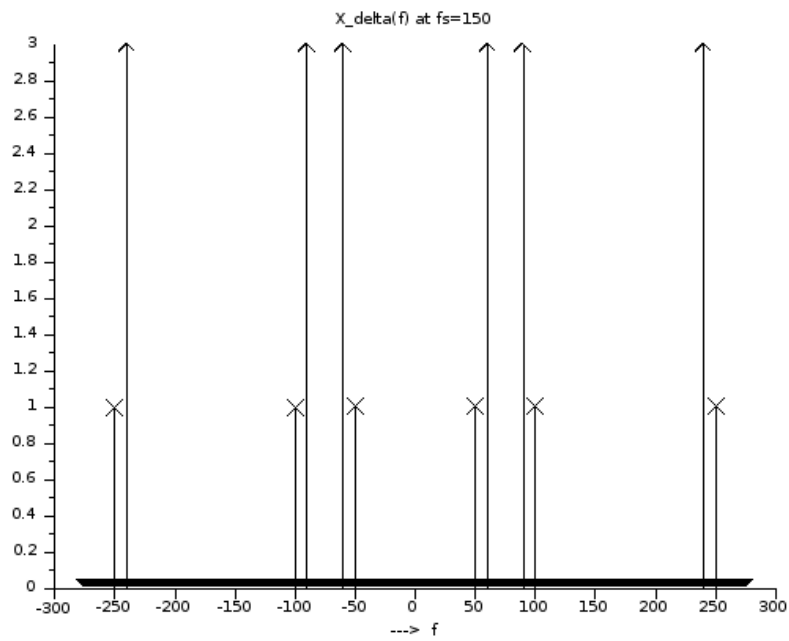


Figure 6.6: LPF output

```
6 syms t;
7 x=12*cos(800*pi*t)*cos((1800*pi*t))^2;
8 disp(x, 'x(t)');
9 disp('Maximum Frequency component present: 2200 Hz')
;
10 fs=2*2200;
11 disp(fs, 'Minimum Sampling Frequency to avoid
distortion is (Hz):');
```

Scilab code Exa 6.8 Minimum Sampling Frequency

```
1 //Scilab Code for Example 6.8 of Signals and systems
by
2 //P.Ramakrishna Rao
3 //Sampling Frequency of BPS
4 clc;
5 clear;
6 disp('Maximum Frequency component present: 25 kHz');
7 fs=2*25000;
8 disp(fs, 'Minimum Sampling Frequency to avoid
distortion is:(Hz)');
```

Chapter 7

Systems

Scilab code Exa 7.1 Properties of System

```
1 //Scilab Code for Example 7.1 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clc;
4 clear;
5 a0=2;
6 a1=2;
7 a2=4;
8 x1=[1,3,5,7];
9 x2=[2,4,6,8];
10 for t=1:4
11     y1(1,t)=a0+a1*x1(t)+a2*(x1(t))^2;
12     y2(1,t)=a0+a1*x2(t)+a2*(x2(t))^2;
13 end
14 b1=2;
15 b2=3;
16 x=b1*x1+b2*x2;
17 disp('y(n) does not depend on past inputs');
18 disp('Hence the system is Static');
19 disp(x,'The input to the system is:');
20 for t=1:4
```



```

21     q(1,t)=a0+a1*x(t)+a2*(x(t))^2;
22 end
23 disp(q, 'This input gives the output:');
24 y=b1*y1+b2*y2;
25 disp(y, 'For the system to be linear the output
    should be:');
26 disp('Hence the system is not linear');

```

Scilab code Exa 7.3 System Properties

```

1 //Scilab Code for Example 7.3 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clc;
4 clear x y1 y y2 q t n;
5 clear;
6 //y(n)=x(n)-x(n-1);
7 disp('y(n) depends upon past inputs also');
8 disp('Output at n=2 depends upon value of x at n=1')
  ;
9 disp('Hence the system is Dynamic');
10 x1=[1,3,5,7,2];
11 x2=[2,4,6,8,3];
12 for n=2:5
13     y1(1,n)=x1(n)-x1(n-1)
14     y2(1,n)=x2(n)-x2(n-1)
15 end
16 b1=2;
17 b2=3;
18 x=b1*x1+b2*x2
19 disp(x, 'The input to the system is:');
20 for n=2:5
21     q(1,n)=x(n)-x(n-1);
22 end
23 y=b1*y1+b2*y2;

```

```

24 disp(q, 'This input gives the output:');
25 disp(y, 'For the system to be linear the output
    should be:');
26 disp('Hence the system is linear');

```

Scilab code Exa 7.4 System Properties

```

1 //Scilab Code for Example 7.4 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clc;
4 clear x y1 y y2 q t n;
5 clear;
6 //y(t)=x(2*t)
7 disp('y(t) depends upon past inputs for t<0');
8 disp('y(t) depends upon future inputs for t>0');
9 disp('Hence the system is Dynamic');
10 x1=[1,3,5,7,2,5,3,9]; //Random Variable
11 x2=[2,4,6,8,2,4,2,1];
12 for t=1:4
13     y1(1,t)=x1(2*t);
14     y2(1,t)=x2(2*t);
15 end
16 b1=2;
17 b2=3;
18 x=b1*x1+b2*x2;
19 disp(x, 'The input to the system is:');
20 for t=1:4
21     q(1,t)=x(2*t);
22 end
23 disp(q, 'This input gives the output:');
24 y=b1*y1+b2*y2;
25 disp(y, 'For the system to be linear the output
    should be:');
26 disp('Hence the system is linear');

```

```

27 disp('For a delay (T) of 2 seconds ');
28 disp('At t=3 seconds: ');
29 t=3;
30 a=x(1,2*t-2);
31 b=y(1,t-2);
32 c=x(1,2*t-4);
33 disp(a, 'x(2t-T): ');
34 disp(b, 'is not equal to y(2t-T): ');
35 disp(c, 'while x(2t-2*T): ');
36 disp('Hence the system is Time variant ');

```

Scilab code Exa 7.5 System Properties

```

1 //Scilab Code for Example 7.5 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clc;
4 clear x y1 y y2 q t n;
5 clear;
6 //y(t)=x(2*n)
7 disp('y(n) depends upon past inputs for n<0');
8 disp('y(n) depends upon future inputs for n>0');
9 disp('Hence the system is Dynamic');
10 x1=[1,3,5,3,2,5,3,9]; //random variable
11 x2=[2,4,6,4,2,4,2,1];
12 for n=1:4
13     y1(1,n)=x1(2*n);
14     y2(1,n)=x2(2*n);
15 end
16 b1=2;
17 b2=3;
18 x=b1*x1+b2*x2;
19 disp(x, 'The input to the system is: ');
20 for n=1:4
21     q(1,n)=x(2*n);

```

```

22 end
23 disp(q, 'This input gives the output:');
24 y=b1*y1+b2*y2;
25 disp(y, 'For the system to be linear the output
    should be:');
26 disp('Hence the system is linear');
27 disp('For a delay (n0) of 2 seconds');
28 disp('At n=3 seconds:');
29 t=3;
30 a=x(1,2*n-2);
31 b=y(1,n-2);
32 c=x(1,2*n-4);
33 disp(a, 'x(2n-n0):');
34 disp(b, 'is not equal to y(2n-n0):');
35 disp(c, 'while x(2n-2*n0):');
36 disp('Hence the system is Time variant');

```

Scilab code Exa 7.6 System Properties

```

1 //Scilab Code for Example 7.6 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clc;
4 clear x y1 y y2 q t n;
5 clear;
6 //y(n)=x(n)-x(n-1);
7 x=[2,4,3,6,7]
8 for n=2:5
9     y(1,n)=x(n)-x(n-1);
10 end
11 disp(y, 'This input gives the output:');
12 disp('For a shift (n0) of 2 seconds');
13 disp('At n=3 seconds:');
14 n=5;
15 b=y(n-2);

```

```

16 a=x(n-2)-x(n-2-1);
17 disp(a, 'x(n-2):');
18 disp(b, 'is equal to y(n-2):');
19 disp('Hence the system is Shift invariant/fixed');

```

Scilab code Exa 7.7 Properties of System

```

1 //Scilab Code for Example 7.7 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clc;
4 clear x y1 y y2 q t n;
5 clear;
6 //y(t)=n*x(n)
7 disp('y(n) depends only upon present inputs');
8 disp('(i) Hence the system is Dynamic');
9 x1=[1,3,5,3,2,5,3,9]; //random variable
10 x2=[2,4,6,4,2,4,2,1];
11 for n=1:4
12     y1(1,n)=n*x1(n);
13     y2(1,n)=n*x2(n);
14 end
15 b1=2;
16 b2=3;
17 x=b1*x1+b2*x2;
18 disp(x, 'The input to the system is:');
19 for n=1:4
20     q(1,n)=n*x(n);
21 end
22 disp(q, 'This input gives the output:');
23 y=b1*y1+b2*y2;
24 disp(y, 'For the system to be linear the output
  should be:');
25 disp('(ii) Hence the system is linear');
26 disp('For a delay (n0) of 2 seconds');

```

```

27 disp('At n=3 seconds:');
28 t=3;
29 a=x(1,n-2);
30 b=y(1,n-2);
31 c=2*x(1,n-2);
32 disp(a,'x(n-n0):');
33 disp(b,'is not equal to y(n-n0):');
34 disp(c,'while (n-n0)*x(n-n0):');
35 disp('(iii) Hence the system is Time variant');

```

Scilab code Exa 7.8 Properties of System

```

1 //Scilab Code for Example 7.8 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clc;
4 clear x y1 y y2 q t n;
5 clear;
6 //y(t)=sum(x(n)),n-2<=n<=n+2
7 x1=[1,3,5,3,0,0,0,0,0,0]; //random variable
8 x2=[2,4,6,4,0,0,0,0,0,0];
9 for n=1:4
10     y1(1,n)=x1(n)+x1(n+1)+x1(n+2)+x1(n+3)+x1(n+4);
11     y2(1,n)=x2(n)+x2(n+1)+x2(n+2)+x2(n+3)+x2(n+4);
12 end
13 b1=2;
14 b2=3;
15 x=b1*x1+b2*x2;
16 disp(x,'The input to the system is:');
17 for n=1:4
18     q(1,n)=x(n)+x(n+1)+x(n+2)+x(n+3)+x(n+4);
19 end
20 disp(q,'This input gives the output:');
21 y=b1*y1+b2*y2;
22 disp(y,'For the system to be linear the output

```

```

    should be:');
23 disp('(i) Hence the system is linear');

```

Scilab code Exa 7.9 Properties of System

```

1 //Scilab Code for Example 7.9 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clc;
4 clear x y1 y y2 q t n;
5 clear;
6 //y(t)=exp|x(n)|
7 x1=[1,3,5,3]; //random variable
8 x2=[2,4,6,4];
9 for n=1:4
10     y1(1,n)=exp(abs(x1(n)));
11     y2(1,n)=exp(abs(x2(n)));
12 end
13 b1=2;
14 b2=3;
15 x=b1*x1+b2*x2;
16 disp(x, 'The input to the system is:');
17 for n=1:4
18     q(1,n)=exp(abs(b1*(x1(n))+b2*(x2(n))));
19 end
20 disp(q, 'This input gives the output:');
21 y=b1*y1+b2*y2;
22 disp(y, 'For the system to be linear the output
    should be:');
23 disp('(ii) Hence the system is not linear');
24 disp('For a delay (n0) of 2 seconds');
25 disp('At n=3 seconds:');
26 n=4;
27 a=exp(abs(x1(n-2)));
28 b=y1(1,n-2);

```

```
29 disp(a, 'e^x(n-n0):');
30 disp(b, 'is equal to y(n-n0):');
31 disp('(iii) Hence the system is Time invariant');
```

Scilab code Exa 7.10 Output of an LTI

```
1 //Scilab Code for Example 7.10 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 disp('h(t)=e^-2*t.u(t)');
6 for w=1:100
7     y(1,w)=integrate('exp(-2*t)', 't', 0, w);
8 end
9 w=1:100;
10 plot(w,y)
11 title('Output Signal y(t)');
12 xlabel('Time');
13 ylabel('Amplitude');
```

Scilab code Exa 7.11 Convolution

```
1 //Scilab Code for Example 7.11 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 clear x y n;
6 x=[0,0,2,0,0];
7 y=[0,0,1,1,0];
```

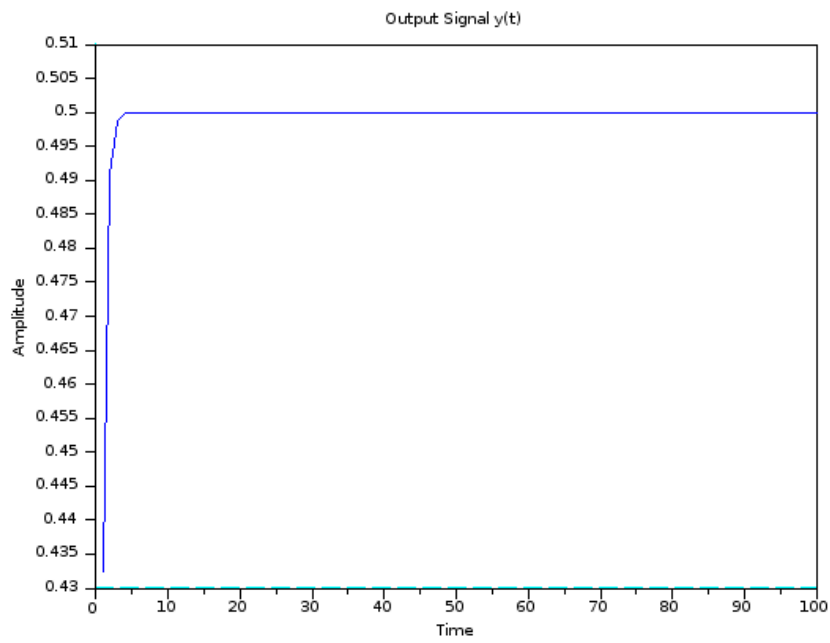



Figure 7.1: Output of an LTI

```

8 n=-2:2;
9 c = gca();
10 c.y_location = "origin";
11 c.x_location = "origin";
12 plot2d2(n,x,2);
13 title('x(t)')
14 xlabel('t')
15 figure(1);
16 n=-2:2;
17 c = gca();
18 c.y_location = "origin";
19 c.x_location = "origin";
20 plot2d2(n,y,5);
21 title('y(t)')
22 xlabel('t')
23 z=conv(x,y);
24 figure(2);
25 n=-3:5;
26 c = gca();
27 c.y_location = "origin";
28 c.x_location = "origin";
29 plot(n,z,2);
30 title('Convolved signal      z(t)')
31 xlabel('t')

```

Scilab code Exa 7.14 Impulse and Step response

```

1 //Scilab Code for Example 7.14 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Plotting the impulse and step responses
4 clc;
5 clear;

```

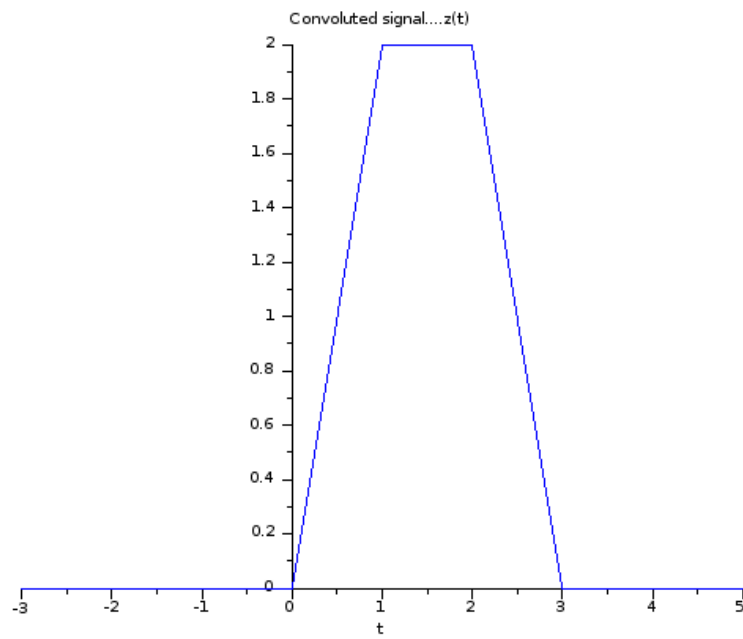


Figure 7.2: Convolution

```

6 syms s t R C;
7 Y1=(1/(R*C))/(s+1/(R*C));
8 disp(Y1,'Laplace Transform Of differential Equation
      is:')
9 y11=ilaplace(Y1,s,t);
10 disp(y11,'The Impulse Response of the System is:');
11 for k=0:10;
12     y1(k+1)=exp(-k);
13 end
14 k=0:10;
15 plot(k,y1);
16 title('System Response to impulse input');
17 Y2=(1/(R*C))/(s^2+s/(R*C));
18 disp(Y2,'Laplace Transform Of differential Equation
      is:')
19 y22=ilaplace(Y2,s,t);
20 disp(y22,'The Step Response of the System is:');
21 for k=0:10;
22     y2(k+1)=1-exp(-k);
23 end
24 figure(1);
25 k=0:10;
26 plot(k,y2);
27 title('System Response to Step input');

```

Scilab code Exa 7.15 Impulse and Step Response

```

1 //Scilab Code for Example 7.15 of Signals and
      systems by
2 //P.Ramakrishna Rao
3 //Plotting the impulse and step responses

```

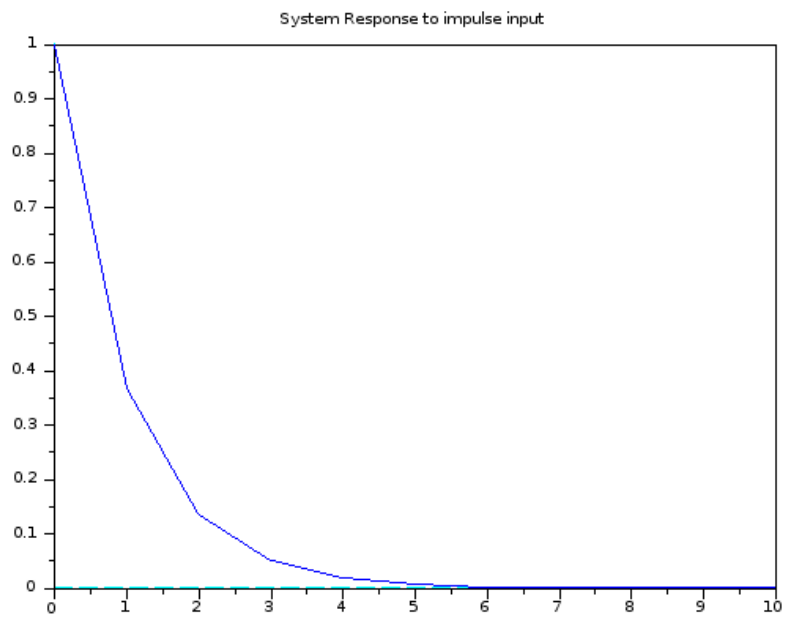


Figure 7.3: Impulse and Step response

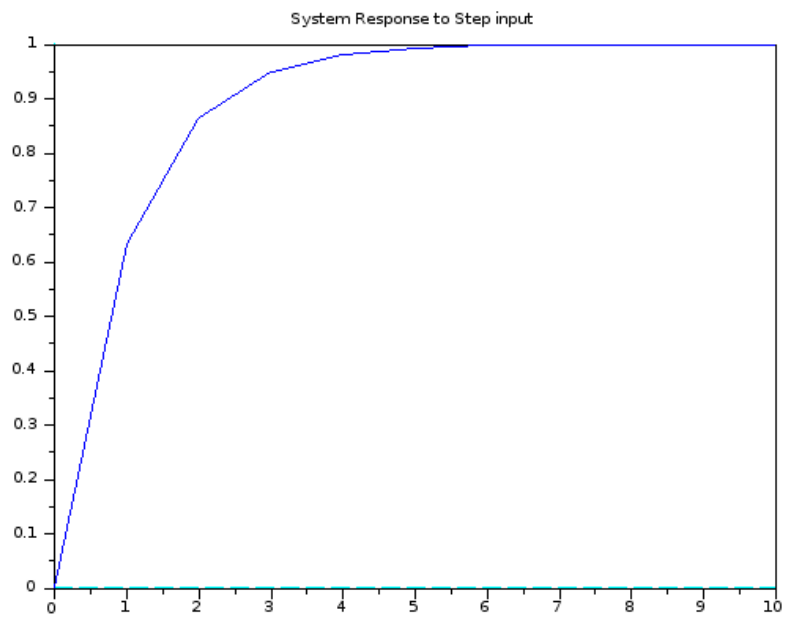


Figure 7.4: Impulse and Step response

```

4  clc;
5  clear;
6  syms s t R L;
7  Y1=(1/s)-(1/(s+(R/L)));
8  disp(Y1,'Laplace Transform Of differential Equation
      is:')
9  y1=ilaplace(Y1,s,t);
10 disp(y1,'The Step Response of the System is:');
11 //Taking R/L=1;
12 for k=0:10;
13     y1(k+1)=1-exp(-k);
14 end
15 k=0:10;
16 plot(k,y1);
17 title('System Response to Step input');
18 Y2=(1/(s+(R/L)));
19 disp(Y2,'Laplace Transform Of differential Equation
      is:')
20 y2=ilaplace(Y2,s,t);
21 disp(y2,'The Impulse Response of the System is:');
22 for k=0:10;
23     y2(k+1)=exp(-k);
24 end
25 figure(1);
26 k=0:10;
27 plot(k,y2);
28 title('System Response to impulse input');

```

Scilab code Exa 7.19 Convolution

```

1 //Scilab Code for Example 7.19 of Signals and

```

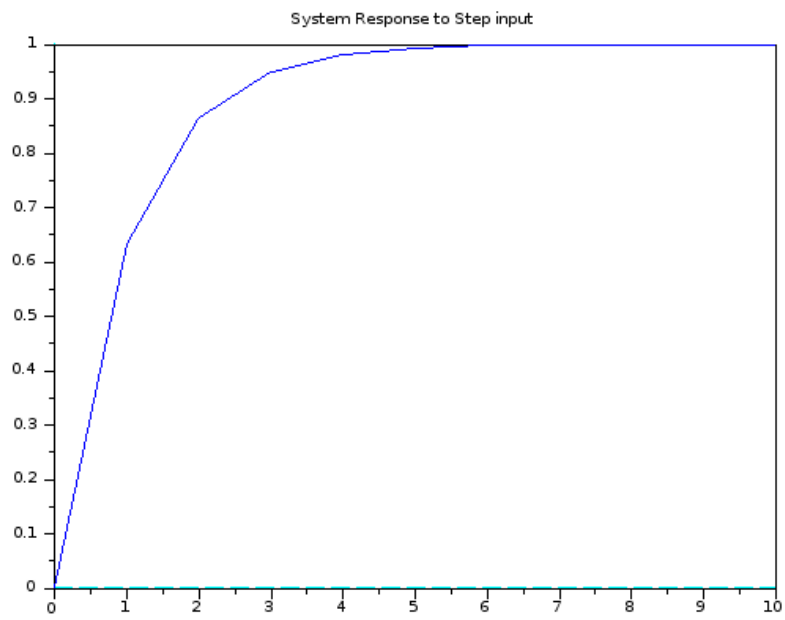


Figure 7.5: Impulse and Step Response

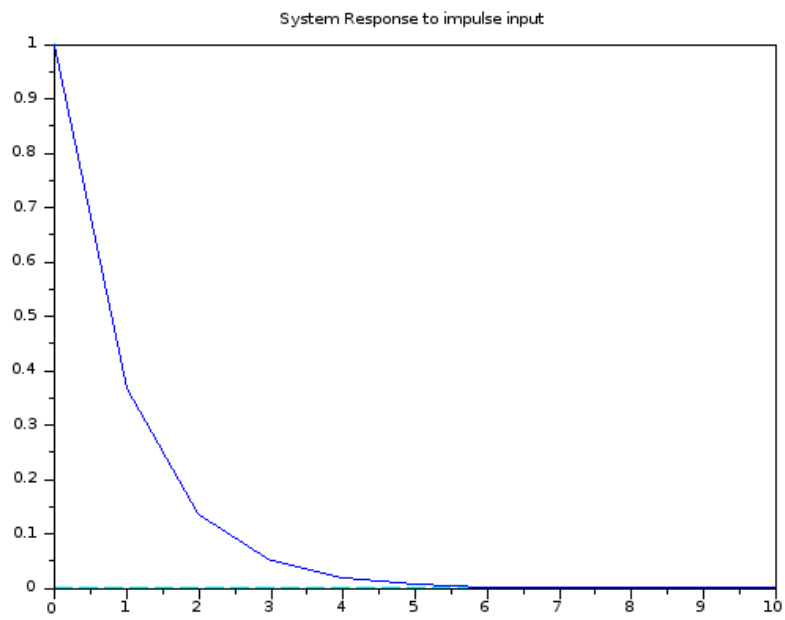


Figure 7.6: Impulse and Step Response

```

        systems by
2 //P.Ramakrishna Rao
3 //Convolution of two signals
4 clc;
5 clear;
6 clear x y n;
7 x=[2,-1,1,0,2];
8 y=[1,0,-1,2];
9 n=-1:3;
10 c = gca();
11 c.y_location = "origin";
12 c.x_location = "origin";
13 plot2d3(n,x,-5);
14 title('x(k)')
15 xlabel('k')
16 figure(1);
17 n=0:3;
18 c = gca();
19 c.y_location = "origin";
20 c.x_location = "origin";
21 plot2d3(n,y,-5);
22 title('y(k)')
23 xlabel('k')
24 z=conv(x,y);
25 figure(2);
26 n=-1:6;
27 c = gca();
28 c.y_location = "origin";
29 c.x_location = "origin";
30 plot2d3(n,z,-5);
31 title('Convolved signal      z(t)')
32 xlabel('t')

```

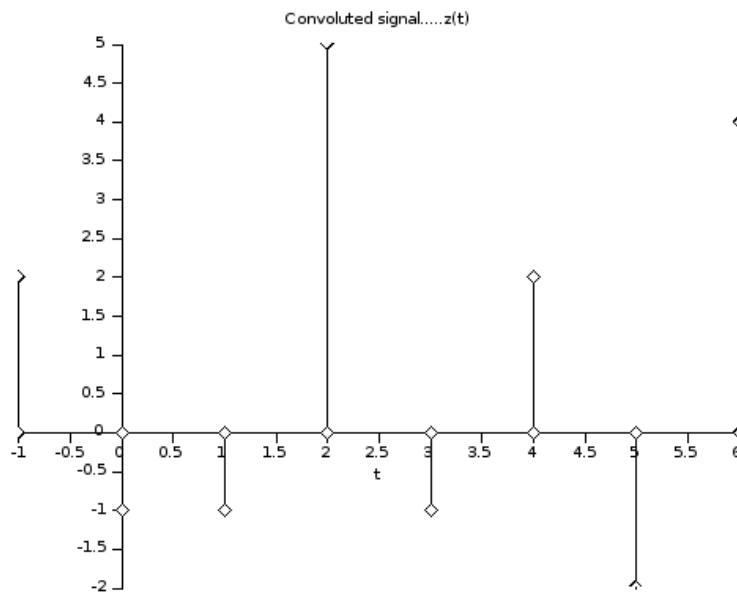


Figure 7.7: Convolution

Scilab code Exa 7.22 Convolution

```
1 //Scilab Code for Example 7.22 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Convolution of two signals
4 clc;
5 clear;
6 clear x y n;
7 for n=0:10;
8     x(n+1)=(3/4)^n*u(n);
9 end
10 c = gca();
11 c.y_location = "origin";
12 c.x_location = "origin";
13 n=0:10;
14 plot2d3(n,x,-4);
15 title('x(n)')
16 xlabel('n')
17 for n=0:10;
18     y(n+1)=u(n);
19 end
20 figure(1);
21 n=0:10;
22 c = gca();
23 c.y_location = "origin";
24 c.x_location = "origin";
25 plot2d3(n,y,-4);
26 title('y(k)')
27 xlabel('k')
28 z=conv(x,y);
29 figure(2);
30 n=0:20;
31 c = gca();
32 c.y_location = "origin";
33 c.x_location = "origin";
34 plot2d3(n,z,-4);
35 title('Convoluted signal      w(t)');
```

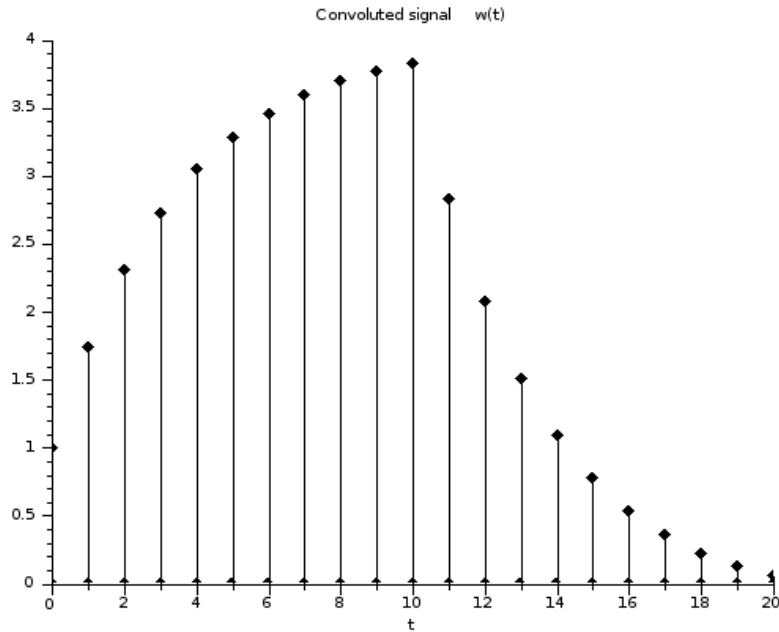


Figure 7.8: Convolution

```

36 xlabel('t');
37 disp(z(1), 'z(0)', z(4), 'z(3)', z(6), 'z(5)', z(11), 'z
(10)');

```

Scilab code Exa 7.23 Convolution

```

1 //Scilab Code for Example 7.23 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //Convolution of two signals
4 clc;

```

```

5 clear;
6 clear x y n;
7 for n=0:10;
8     x(n+1)=(0.5)^n*u(n-2);
9 end
10 c = gca();
11 c.y_location = "origin";
12 c.x_location = "origin";
13 n=0:10;
14 plot2d3(n,x,-4);
15 title('x(n)')
16 xlabel('n')
17 for n=0:10;
18     y(n+1)=u(n);
19 end
20 figure(1);
21 n=0:10;
22 c = gca();
23 c.y_location = "origin";
24 c.x_location = "origin";
25 plot2d3(n,y,-4);
26 title('y(k)')
27 xlabel('k')
28 z=conv(x,y);
29 figure(2);
30 n=0:20;
31 c = gca();
32 c.y_location = "origin";
33 c.x_location = "origin";
34 plot2d3(n,z,-4);
35 title('Convolved signal      z(t)')
36 xlabel('t')

```

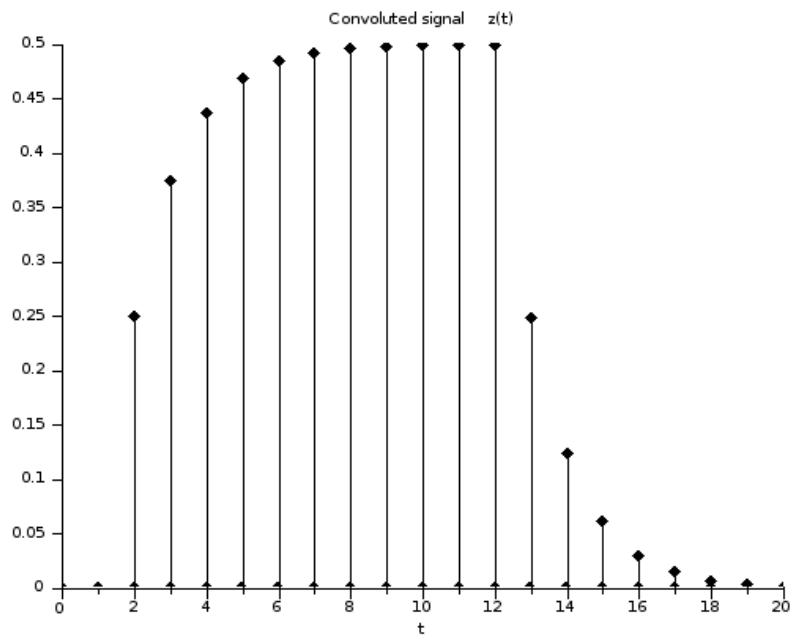


Figure 7.9: Convolution

Scilab code Exa 7.24 Step Response

```
1 //Scilab Code for Example 7.24 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Plotting the step response
4 clc;
5 clear;
6 syms z n;
7 y1=2*(1-0.5^n);
8 disp(y1*'u(n)', 'The step Response of the System is:');
9 for n=0:10;
10     q(n+1)=2*(1-0.5^n);
11 end
12 n=0:10;
13 plot(n,q);
14 title('Step Response    g(n)');
15 xlabel('n-->');
```

Scilab code Exa 7.25 Step response

```
1 //Scilab Code for Example 7.25 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Plotting the impulse and step responses
4 clc;
5 clear;
6 syms z a n;
7 Y1=(2*z/(z-a));
8 disp(Y1, 'Z Transform Of differential Equation is:');
9 y12=2*a^n;
10 disp(y12, 'The Unit Sample Response of the System is:');
```

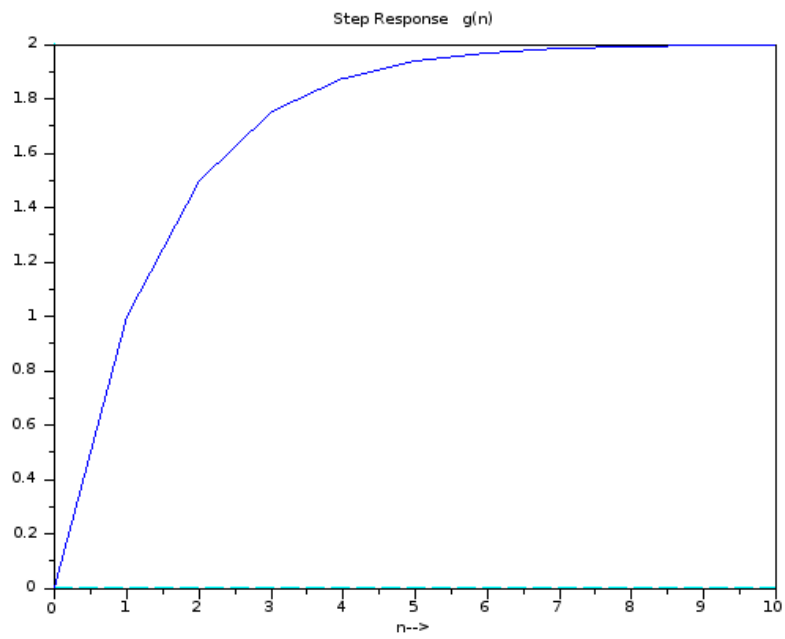



Figure 7.10: Step Response

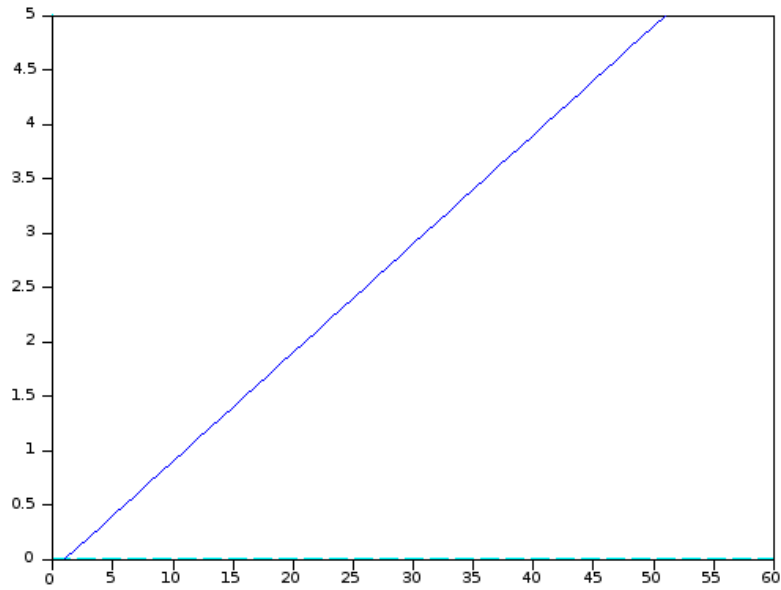


Figure 7.11: Step response

```

    ');
11 y2=2*(a^n-1)/(a-1);
12 disp(y2,'The Step Response of the System is:');
13 q=1;
14 a=0.5;
15 for k=0:0.1:5;
16 y2(q)=2*((a^k-1)/(a-1));
17 q=q+1;
18 end
19 k=0:0.1:5;
20 plot(k,y2);

```

Scilab code Exa 7.33 Magnitude and Phase Response

```
1 //Scilab Code for Example 7.33 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Plotting the magnitude and phase responses
4 clc;
5 clear;
6 T=1;
7 n=1;
8 for w=0:0.1:20;
9     hmag(n)=2*sin(w*T/2);
10    n=n+1;
11 end
12 n=1;
13 for w=0:0.1:20;
14    hphase(n)=%pi/2-(w*T/2);
15    n=n+1;
16 end
17 //Magnitude plot
18 w=0:0.1:20;
19 plot(w,hmag);
20 title('Magnitude Plot');
21 xlabel('w');
22 ylabel('|H(ejw)|');
23 figure(1);
24 //Phase Plot
25 w=0:0.1:20;
26 plot(w,hphase);
27 title('Phase Plot');
28 xlabel('w');
29 ylabel('theta(wT)');
```

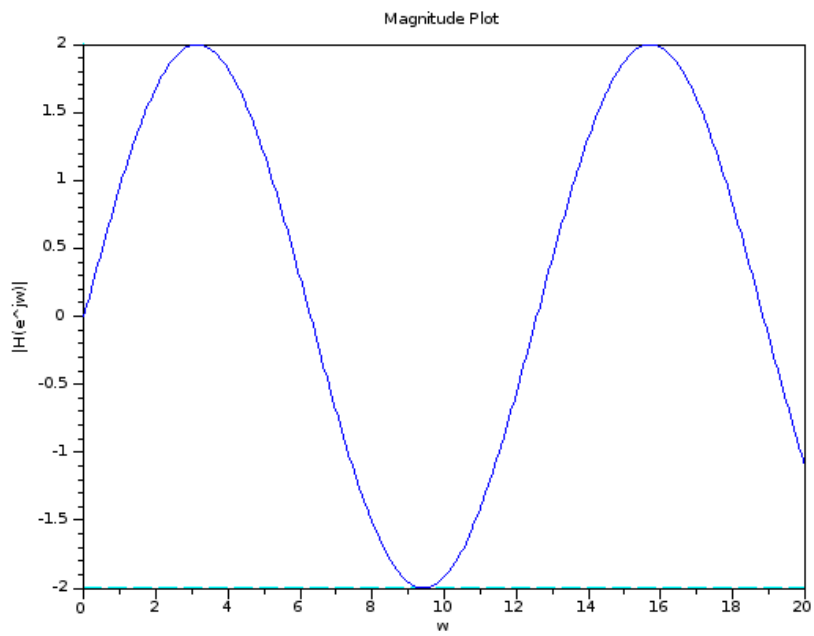


Figure 7.12: Magnitude and Phase Response

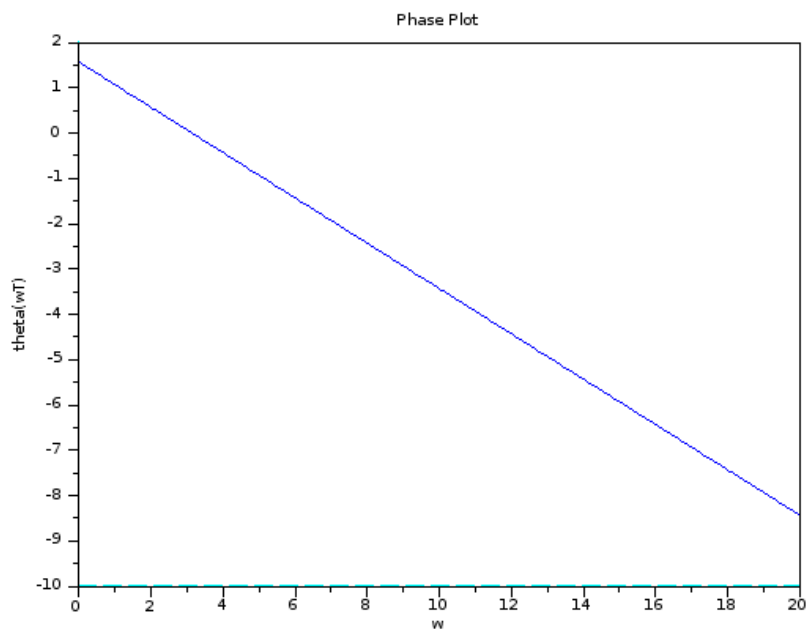


Figure 7.13: Magnitude and Phase Response

Chapter 8

Systems Time and Frequency Domain Analysis

Scilab code Exa 8.1 Magnitude and Phase Responses

```
1 //Scilab Code for Example 8.1 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //First Order causal LTI system
4 clear;
5 clc;
6 syms s t;
7 Y=1/(s+6);
8 h=exp(-6*t);
9 H=laplace(h,t,'jw');
10 disp(H,"H(w)=");
11 n=1;
12 for w=-5*2*%pi:0.01:5*2*%pi
13     Hmag(n)=1/sqrt(36+w^2)
14     Hphs(n)=-atan(w/6);
15     n=n+1;
16 end
17 w=-5*2*%pi:0.01:5*2*%pi;
18 plot(w,Hmag);
```

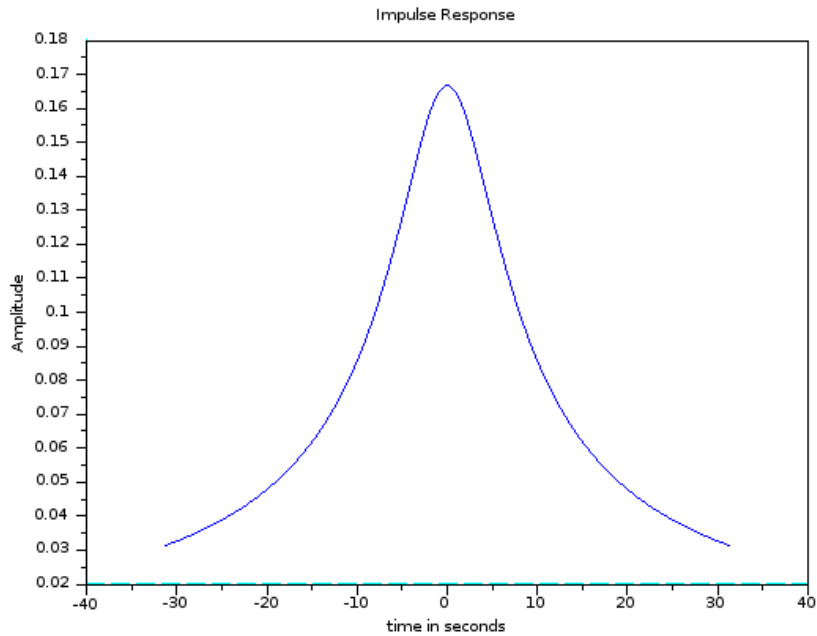


Figure 8.1: Magnitude and Phase Responses

```

19 title('Impulse Response');
20 ylabel('Amplitude');
21 xlabel('time in seconds');
22 figure(1);
23 w=-5*2*%pi:0.01:5*2*%pi;
24 plot(w,Hphs);
25 title('Phase Response');
26 ylabel('theta(w)');
27 xlabel('w in radians/second');

```

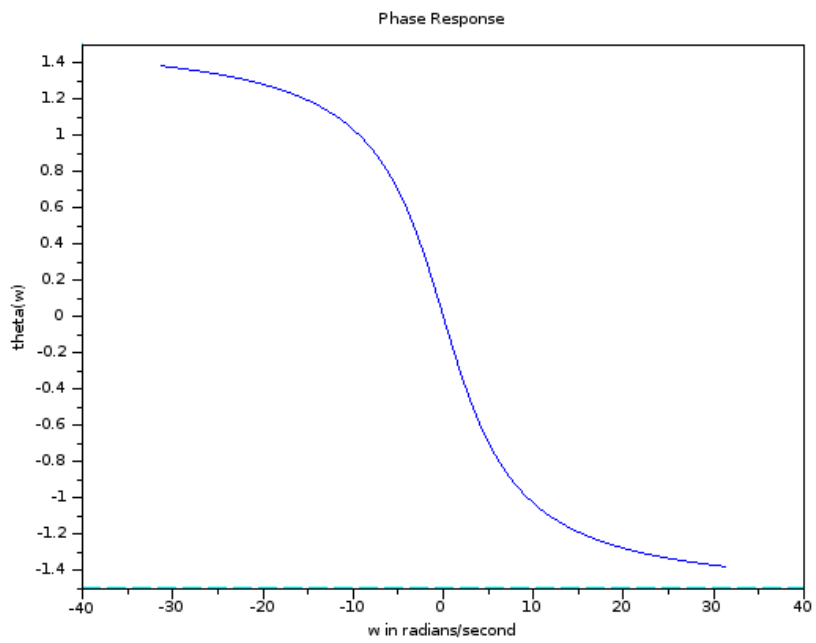


Figure 8.2: Magnitude and Phase Responses

Scilab code Exa 8.2 Transfer Function of system

```
1 //Scilab Code for Example 8.2 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Second order system
4 clear;
5 clc;
6 syms t;
7 L=1;
8 R=1.2;
9 C=10^-6;
10 Rcr=2*sqrt(L/C);
11 eta=R/Rcr;
12 disp(eta, 'Damping Ratio=', Rcr, ' Critical Resistance (
    ohm) ', '(i) ');
13 Wn=1/sqrt(L*C);
14 disp(Wn, 'Undamped Natural Frequency(Hz) ', '(ii) ');
15 n=1;
16 h=(1250*sin(800*t))*exp(-600*t);
17 H=laplace(h,t, 'jW');
18 for t=0:0.1:2;
19     h(n)=(1250*sin(800*t))*exp(-600*t); //Impulse
      Response
20     n=n+1;
21 end
22 t=0:0.1:2;
23 plot(t,h);
24 title('Impulse Response');
25 ylabel('Amplitude');
26 xlabel('time in seconds');
27 disp(H, '(iv) Transfer Function(H(jw)):');
```

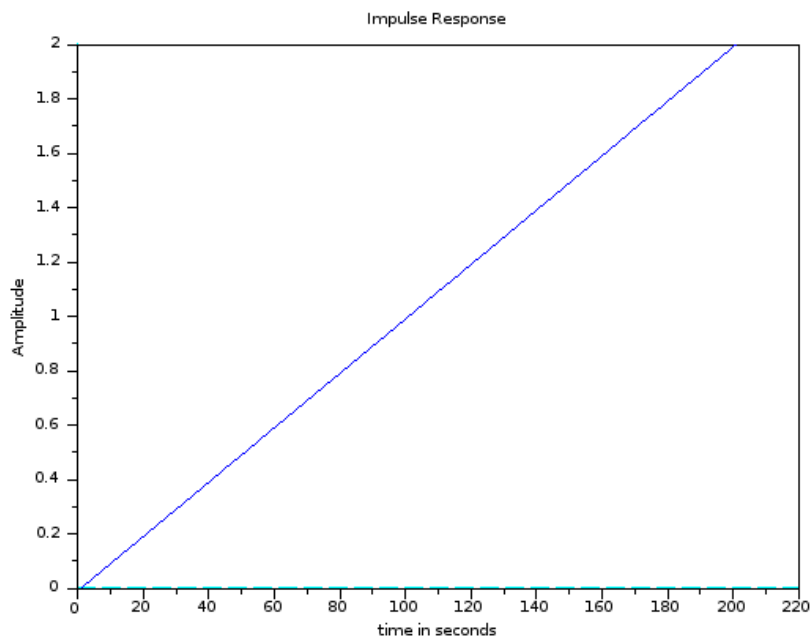


Figure 8.3: Transfer Function of system

Scilab code Exa 8.3 Magnitude and Phase Responses

```
1 //Scilab Code for Example 8.3 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms n w;
6 h=(0.3)^n;
7 disp(h,'The impulse response of the system is:');
8 H=1/(1-(0.3*exp(-%i*w)));
9 disp(H,'The Transfer function of the system is:');
10 n=1;
11 for w=-%pi:0.1:%pi
12     Hmag(n)=1/sqrt(1.09-(0.6*cos(w)));
13     Hphs(n)=-atan(0.3*sin(w)/(1-0.3*cos(w)));
14     n=n+1;
15 end
16 w=-%pi:0.1:%pi;
17 c = gca();
18 c.y_location = "origin";
19 c.x_location = "origin";
20 c.thickness=2;
21 plot(w,Hmag);
22 title('Magnitude Sketch');
23 ylabel('Amplitude');
24 xlabel('W in radians');
25 figure(1);
26 w=-%pi:0.1:%pi;
27 c = gca();
28 c.y_location = "origin";
29 c.x_location = "origin";
30 c.thickness=2;
31 plot(w,Hphs);
32 title('Phase Response');
33 ylabel('theta(w)');
34 xlabel('W in radians')
```

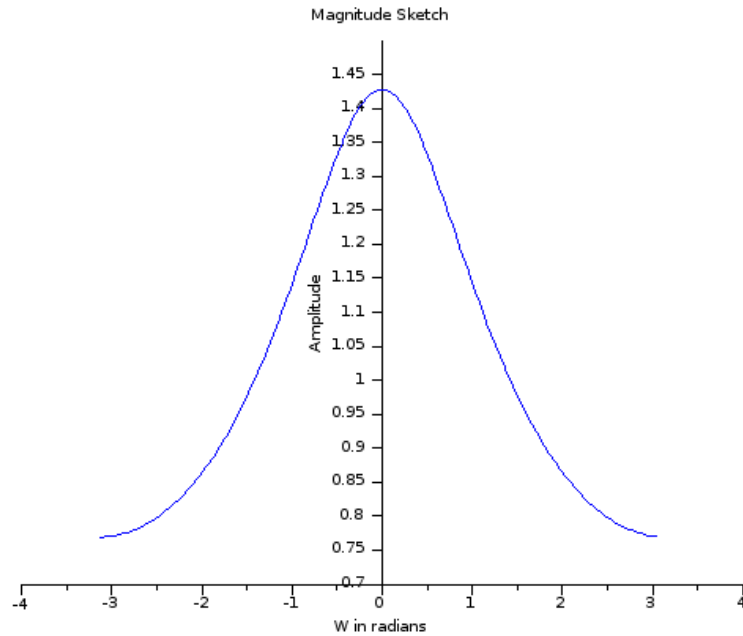


Figure 8.4: Magnitude and Phase Responses

Scilab code Exa 8.4 system response

```

1 //Scilab Code for Example 8.4 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Second Order System
4 clear;
5 clc;
```

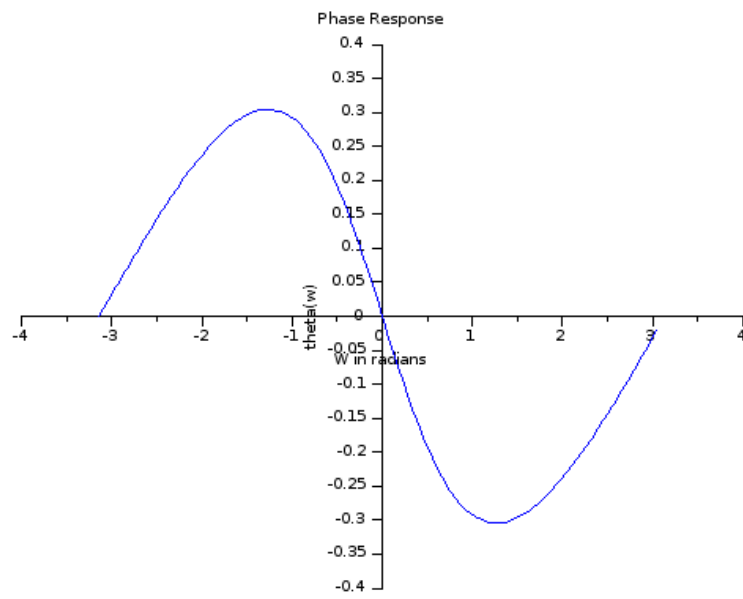


Figure 8.5: Magnitude and Phase Responses

```

6 z = %z;
7 syms n z1;
8 X = (z-1)/((z+(1/2))*(z-(1/5)))
9 disp(X, '(i) System Function is:');
10 X1 = denom(X);
11 zp = roots(X1);
12 X1 = (z1-1)/((z1+(1/2))*(z1-(1/5)));
13 F1 = X1*(z1^(n-1))*(z1-zp(1))
14 F2 = X1*(z1^(n-1))*(z1-zp(2))
15 h1 = limit(F1, z1, zp(1))
16 h2 = limit(F2, z1, zp(2))
17 h = h1+h2;
18 disp('(iv) Unit sample response sequence:');
19 disp(h*"u(n)", 'h[n]=');
20 for n=0:10;
21     w(n+1)=(- 5.714*(0.2)^n- 4.285*(- 0.5^n));
22 end
23 n=0:10;
24 plot(n,w);
25 title('(iii) Unit Sample Response Sequence');
26 disp('where z=e^jw ', X, '(i) Transfer Function is:');

```

Scilab code Exa 8.5 System function

```

1 //Scilab Code for Example 8.5 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Second Order System
4 clear;
5 clc;
6 z = %z;
7 syms n z1;
8 X = z^2/(z-0.5)^2

```

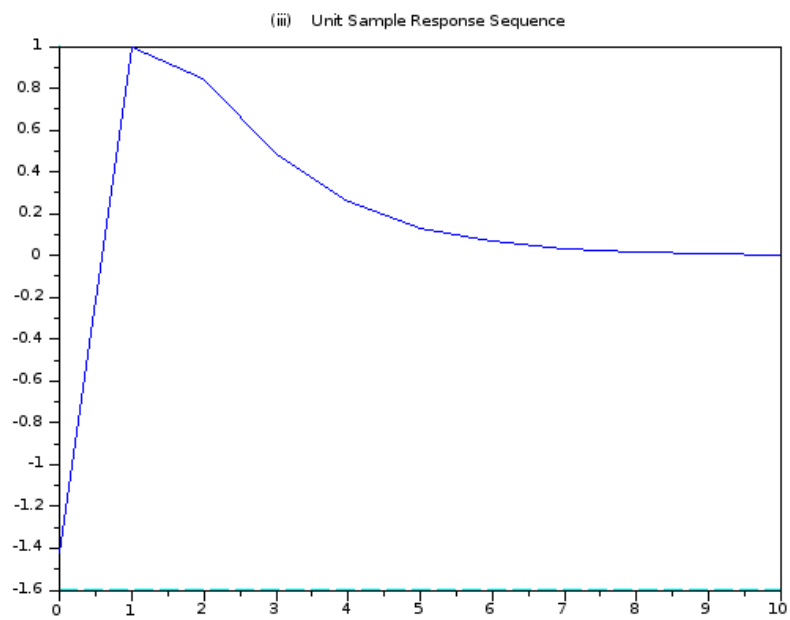


Figure 8.6: system response

```

 9 disp(X, '(i) System Function is:');
10 X1 = denom(X);
11 zp = roots(X1);
12 X1 = z1^2/(z1-0.5)^2;
13 F1 = X1*(z1^(n-1))*(z1-0.5)^2;
14 h1 = limit(F1,z1,zp(1));
15 h = h1;
16 disp('(iii) Unit step response sequence:');
17 disp(h*'u(n)', 'h[n]=');
18 for n=0:20;
19     w1(n+1)=1-(0.5)^n;
20 end
21 n=0:20;
22 plot(n,w1);
23 title('(ii) Unit Sample Response Sequence');
24 disp('where z=e^jw ',X, '(iv) Transfer Function (H(e^
    jw)) is:');

```

Scilab code Exa 8.8 Transfer function

```

1 //Scilab Code for Example 8.8 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms R L C s I;
6 X=I*(R+1/(C*s));
7 Y=R*I;
8 Z=Y/X;
9 disp(Z, '(a) RC High pass Filter: H(s)');
10 X=I*(L*s+1/(C*s));
11 Y=I/(C*s);
12 Z=Y/X;

```

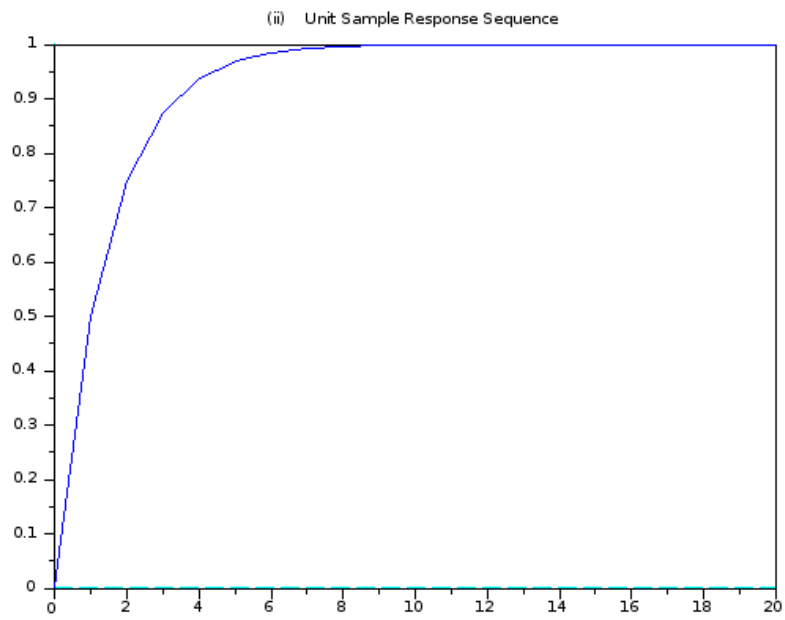



Figure 8.7: System function

```
13 disp(Z, '(b) LC Low pass Filter: H(s)');
```

Scilab code Exa 8.9 Transfer function by block diagram reduction

```
1 //Scilab Code for Example 8.9 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms G1 G2 G3 G4 H1 H2 s;
6 G1=s/(s+5);
7 G2=10/(s+10);
8 H1=s;
9 H2=1/s;
10 disp('(i) Shifting the take off point');
11 H1=H1/G2;
12 disp('(ii) Solving second Feedback Loop ');
13 G3=H1-1;
14 disp('(iii) Solving first Feedback Loop ');
15 G4=G2/(1+G2*H2);
16 Z=G1*G3*G4;
17 disp(Z, '(iv) The Transfer Function is:');
```

Scilab code Exa 8.10 Realization Structure

```
1 //Scilab Code for Example 8.10 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms s1;
6 s=%s;
7 H=poly(0, 's');
```

```

8 H=s^3/(s^3+6*s^2++11*s+6);
9 H1=denom(H);
10 a=roots(H1);
11 H2=5*(s1^3/(s1-a(1)))*(s1/(s1-a(2)))*(s1/(s1-a(3)));
12 disp(H2, '(ii) Cascade Form: H(s)');
13 h1=s1^3*(s1-a(1))/((s1+1)*(s1+2)*(s1+3));
14 h2=s1^3*(s1-a(2))/((s1+1)*(s1+2)*(s1+3));
15 h3=s1^3*(s1-a(3))/((s1+1)*(s1+2)*(s1+3));
16 A = limit(h1,s1,a(1))/(s-a(1));
17 B = limit(h2,s1,a(2))/(s-a(2));
18 C = limit(h3,s1,a(3))/(s-a(3));
19 Z=5*(1+A+B+C);
20 disp(Z, '(iii) Parallel Form Realization: H(s)');

```

Scilab code Exa 8.12 Transfer function Realization

```

1 //Scilab Code for Example 8.12 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Second Order LSI system
4 z = %z;
5 s = %s;
6 X=z^-1/(0.7*z^-2-3.7*z^-1+3);
7 [A]=pfss(z^-1/(0.7*z^-2-3.7*z^-1+3))
8 x1 = horner(A(1),z)
9 x2 = horner(A(2),z)
10 q=denom(X);
11 a=roots(q)
12 H=[x1 x2 z/3];
13 disp(H, '(a) H(z)=');
14 clear z;
15 syms z n;
16 F1 = ((1/6)*(z/(z-1))*z^(n-1)*(z-a(1)))
17 F2 = ((0.7/54)*(z/(z-0.233333))*z^(n-1)*(z-a(2)))
18 w1=limit(F1,z,a(1))

```

```

19 w2=limit(F2,z,a(2))
20 w=w1+w2;
21 disp(w, '(b) h(n))=');
22 z=%z;
23 x11=z*x1/3
24 x12=z*x2/3
25 disp(x11, '(c) Parallel realization: H1(z)');
26 disp(x12, ' H2(z)')

```

Scilab code Exa 8.13 Parallel form realization

```

1 //Scilab Code for Example 8.13 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 clear;
4 clc;
5 syms z1;
6 z=%z;
7 H=poly(0, 'z ');
8 H=z*(z+0.5)*(z+0.25)/((z-0.5)*(z-0.25)*(z-0.125));
9 H1=denom(H);
10 a=roots(H1)
11 h1=(z1+0.5)*(z1+0.25)/((z1-0.5)*(z1-0.25)*(z1-0.125)
   )*(z1-a(1));
12 h2=(z1+0.5)*(z1+0.25)/((z1-0.5)*(z1-0.25)*(z1-0.125)
   )*(z1-a(2));
13 h3=(z1+0.5)*(z1+0.25)/((z1-0.5)*(z1-0.25)*(z1-0.125)
   )*(z1-a(3));
14 A=z*limit(h1,z1,a(1))/(z-a(1));
15 B=z*limit(h2,z1,a(2))/(z-a(2));
16 C=z*limit(h3,z1,a(3))/(z-a(3));
17 Z=A+B+C;
18 disp(Z, 'Parallel Form Realisation: H(z)');

```

Chapter 9

Convolution and Correlation Continuous Time Signals

Scilab code Exa 9.1 Convolution

```
1 //Scilab Code for Example 9.1 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Convolution of two signals
4 clc;
5 n=1;
6 for t=-10:0.1:10
7     x(n)=5*cos(t);
8     y(n)=2*exp(-abs(t));
9     n=n+1;
10 end
11 z=conv(x,y);
12 t2=-20:0.1:20;
13 plot2d(t2,z);
14 title('Output signal');
15 xlabel('Time t—>');
```

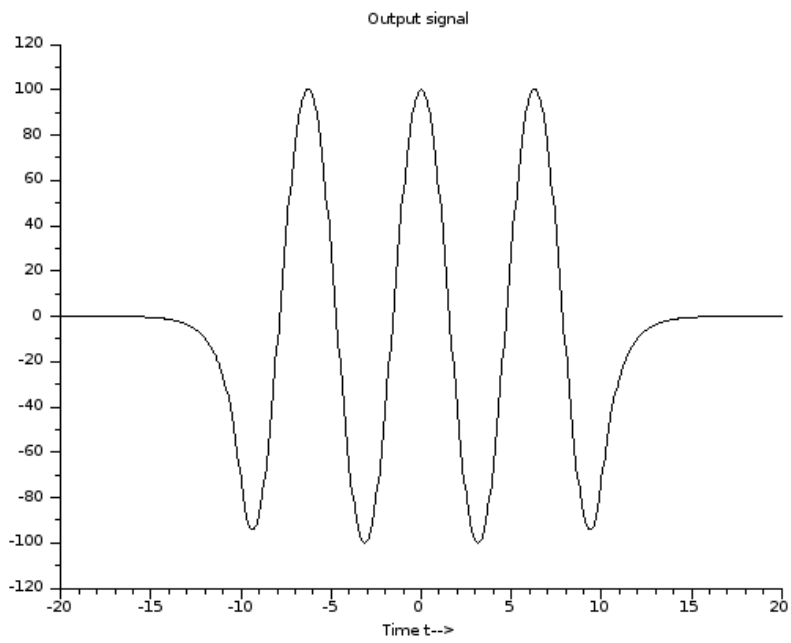


Figure 9.1: Convolution

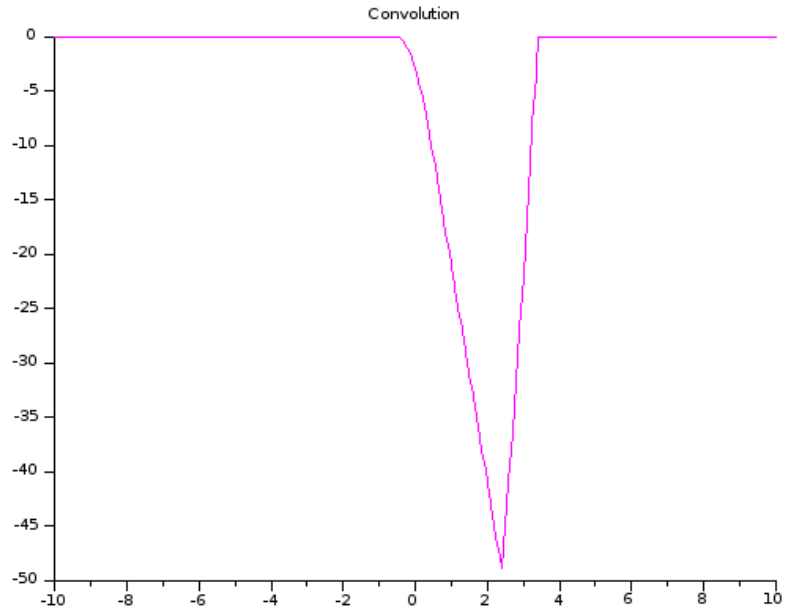


Figure 9.2: Convolution

Scilab code Exa 9.3 Convolution

```

1 //Scilab Code for Example 9.3 of Signals and systems
  by
2 //P.Ramakrishna Rao
3 //Convolution of two signals
4 clc;
5 clear;
6 n=1;
7 for t=-5:0.1:5

```

```

8      x(n)=-2*u(t-0.5)+2*u(t-1.5);
9      y(n)=(t+1)*u(t+1)-(t+1)*u(t-2);
10     n=n+1;
11 end
12 z=conv(x,y);
13 t=-5:0.1:5;
14 plot2d2(t,x,5);
15 title('x(t)');
16 figure(1);
17 t=-5:0.1:5;
18 plot2d(t,y,2);
19 title('y(t)');
20 figure(2);
21 t=-10:0.1:10;
22 plot2d(t,z,6);
23 title('Convolution');

```

Scilab code Exa 9.4 Autocorrelation

```

1 //Scilab Code for Example 9.4 of Signals and systems
   by
2 //P.Ramakrishna Rao
3 clc;
4 clear;
5 T=1;
6 n=1;
7 for t=-T/2:0.01:T/2;
8     x(n)=10*cos(%pi*t/T);
9     n=n+1;
10 end
11 t=-T/2:0.01:T/2;
12 plot(t,x);
13 title('x(t)');

```

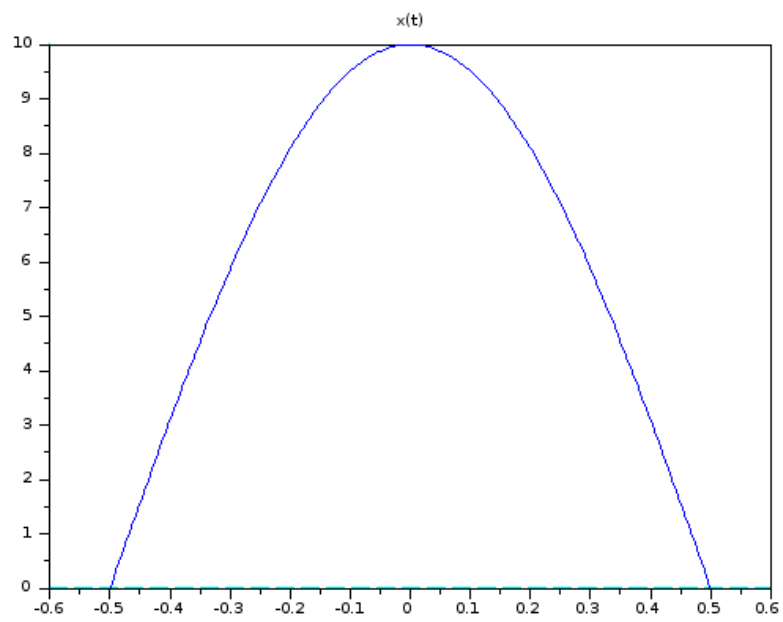



Figure 9.3: Autocorrelation

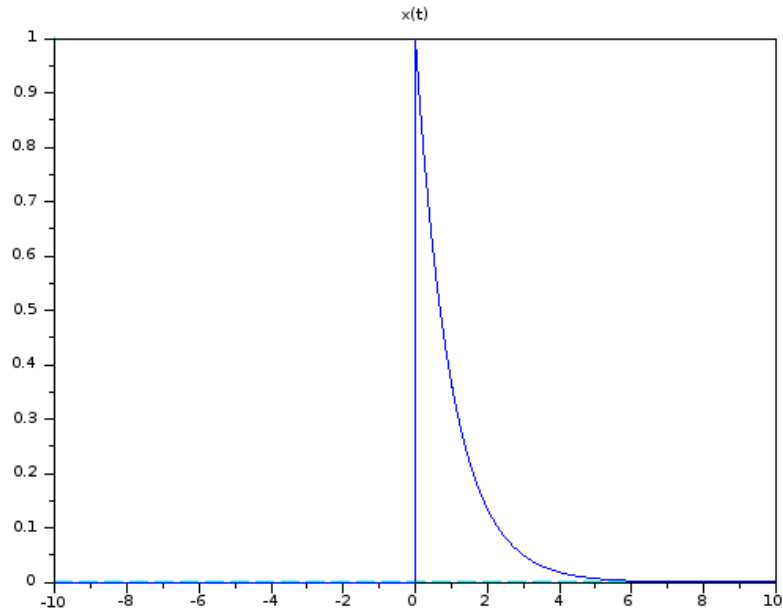


Figure 9.4: Autocorrelation

```

14 disp('Rxx(0)=Energy of signal');
15 Rxx=integrate('50*(1+cos(2*pi*t/T))','t',-T/2,T/2);
16 disp(Rxx,'Rxx(0)=');

```

Scilab code Exa 9.5 Autocorrelation

```

1 //Scilab Code for Example 9.5 of Signals and systems
  by
2 //P.Ramakrishna Rao

```

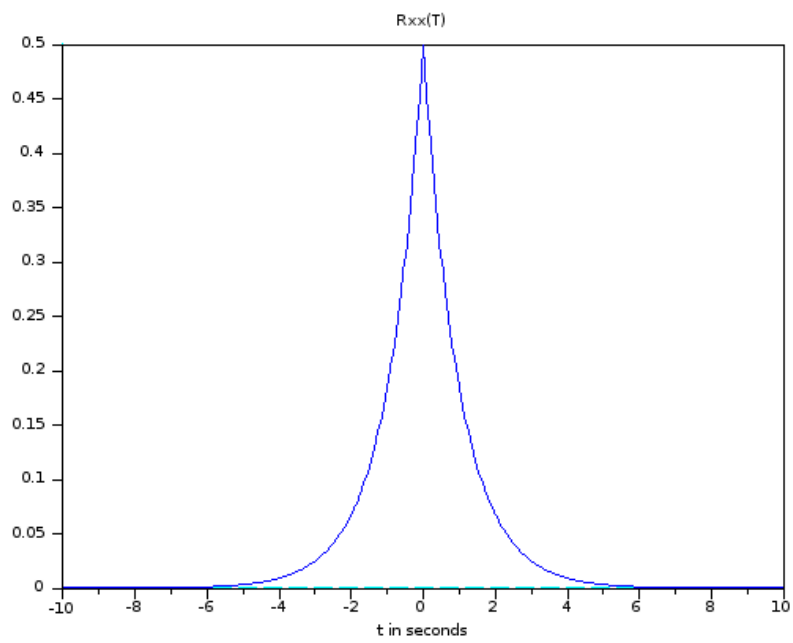


Figure 9.5: Autocorrelation

```

3  clc;
4  clear;
5  clear z x y t T;
6  n=1;
7  for t=-10:0.01:10;
8      x(n)=exp(-t)*u(t);
9      n=n+1;
10 end
11 t=-10:0.01:10;
12 plot(t,x);
13 title('x(t)');
14 n=1;
15 for T=-10:0.1:0;
16     z(n)=integrate('exp(-t)*exp(T-t)', 't', 0, 1000);
17     n=n+1;
18 end
19 n=1;
20 for T=0.1:0.1:10;
21     z(n+101)=integrate('exp(-t)*exp(T-t)', 't', T
22         , 1000);
23     n=n+1;
24 end
25 figure(1);
26 T=-10:0.1:10;
27 plot(T,z);
28 title('Rxx(T)');
29 xlabel('t in seconds');

```

Scilab code Exa 9.6 Autocorrelation

```
1 //Scilab Code for Example 9.6 of Signals and systems
```

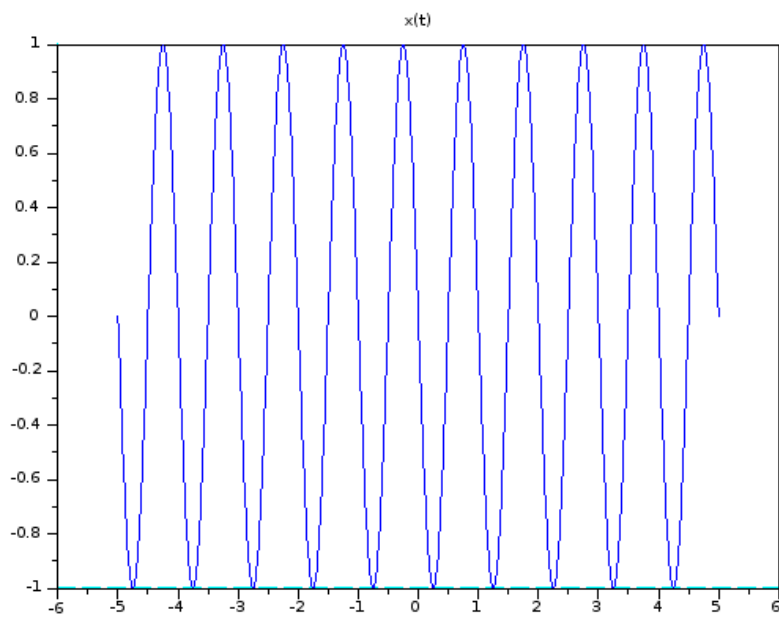


Figure 9.6: Autocorrelation

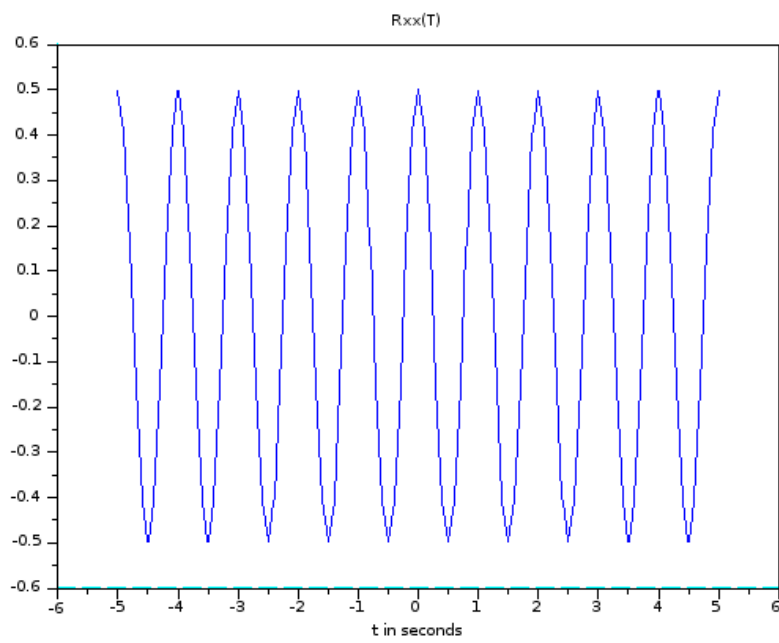


Figure 9.7: Autocorrelation

```

        by
2 //P.Ramakrishna Rao
3 clc;
4 clear;
5 clear z x y t T;
6 A=1;
7 n=1;
8 wo=2*%pi;
9 theta=%pi/2;
10 for t=-5:0.01:5;
11     x(n)=A*cos(wo*t+theta);
12     n=n+1;
13 end
14 t=-5:0.01:5;
15 plot(t,x);
16 title('x(t)');
17 n=1;
18 for T=-5:0.1:5;
19     z(n)=(A^2/2)*cos(wo*T);
20     n=n+1;
21 end
22 figure(1);
23 T=-5:0.1:5;
24 plot(T,z);
25 title('Rxx(T)');
26 xlabel('t in seconds')

```

Scilab code Exa 9.7 Autocorrelation

```

1 //Scilab Code for Example 9.7 of Signals and systems
  by

```

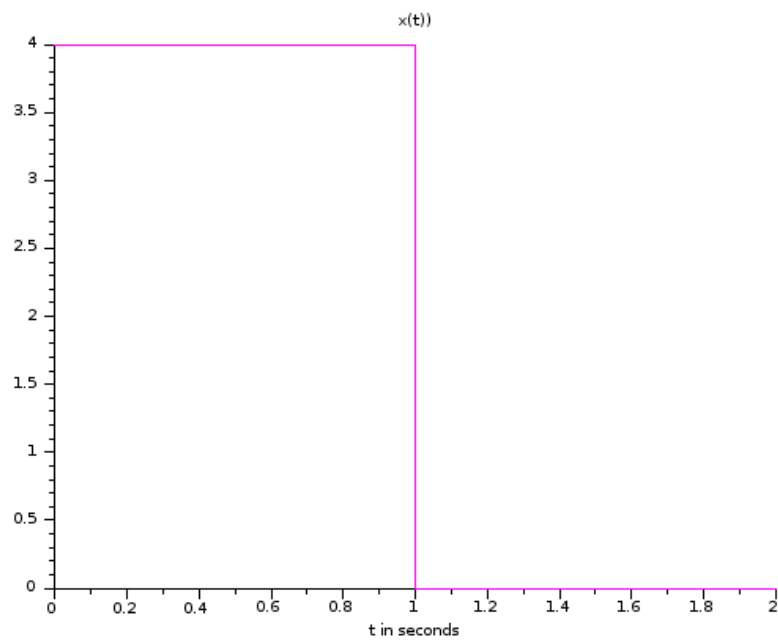


Figure 9.8: Autocorrelation

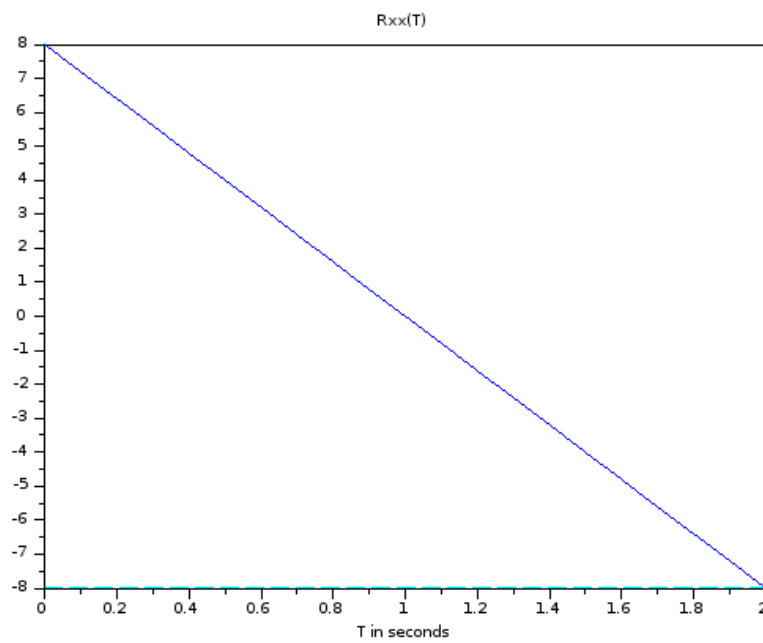


Figure 9.9: Autocorrelation

```

2 //P.Ramakrishna Rao
3 clc;
4 clear;
5 clear z x y t T A;
6 A=4;
7 T0=2;
8 n=1;
9 for T=0:0.1:T0;
10     if T<T0/2 then
11         x(n)=A;
12     else x(n)=0;
13         end
14     n=n+1;
15 end
16 T=0:0.1:T0;
17 plot2d2(T,x,6);
18 title('x(t)');
19 xlabel('t in seconds');
20 n=1;
21 for T=0:0.1:T0;
22     z(n)=(1/T0)*integrate('A^2','t',T,T0/2);
23     n=n+1;
24 end
25 T=0:0.1:T0;
26 figure(1);
27 plot(T,z);
28 title('Rxx(T)');
29 xlabel('T in seconds');

```

Scilab code Exa 9.10 Autocorrelation Function

```

1 //Scilab Code for Example 9.10 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //Maximum Value of Auto-Correlation Function

```

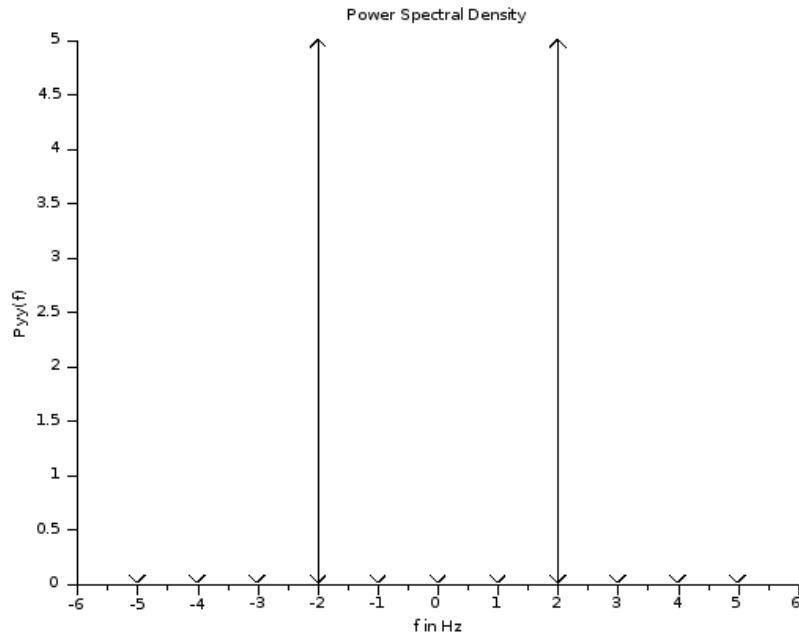


Figure 9.10: PSD

```

4 clc;
5 clear;
6 disp('Maximum Value of ACF=Rxx(0)');
7 x=200*integrate('exp(-2*t)', 't', 0, 1000);
8 disp(x, 'Energy in x(t)=');

```

Scilab code Exa 9.11 PSD

```

1 //Scilab Code for Example 9.11 of Signals and
  systems by
2 //P.Ramakrishna Rao

```

```

3 //Power Spectral Density
4 clear;
5 clc;
6 clear f n;
7 fo=2;
8 n=1;
9 RC=1/(2*%pi*10^3);
10 w=4*%pi*10^3
11 H=1/(1+%i*w*RC);
12 disp(H, 'H(f) ');
13 X=(abs(H))^2;
14 for f=-5:5
15     Pxx(n)=25*[delta(f-fo)+delta(f+fo)];
16     Pyy(n)=X*Pxx(n);
17     n=n+1;
18 end
19 disp(Pxx, 'Pxx=');
20 disp(Pyy, 'Pyy=');
21 f=-5:5;
22 plot2d3(f,Pyy,-2);
23 title('Power Spectral Density');
24 ylabel('Pyy(f)');
25 xlabel('f in Hz');

```

Chapter 10

Discrete Time Convolution and Correlation

Scilab code Exa 10.1 Cross Correlation

```
1 //Scilab Code for Example 10.1 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Cross Correlation
4 clear;
5 clc;
6 x=[2,-1,1,0,2];
7 y=[0,1,0,-1,2];
8 //computation of cross correlation sequence;
9 n1 = max(size(y))-1;
10 n2 = max(size(x))-1;
11 r = xcorr(x,y,n1);
12 n=-4:4;
13 a=gca();
14 a.x_location="origin";
15 a.y_location="origin";
16 plot2d3(n,r,-9);
```

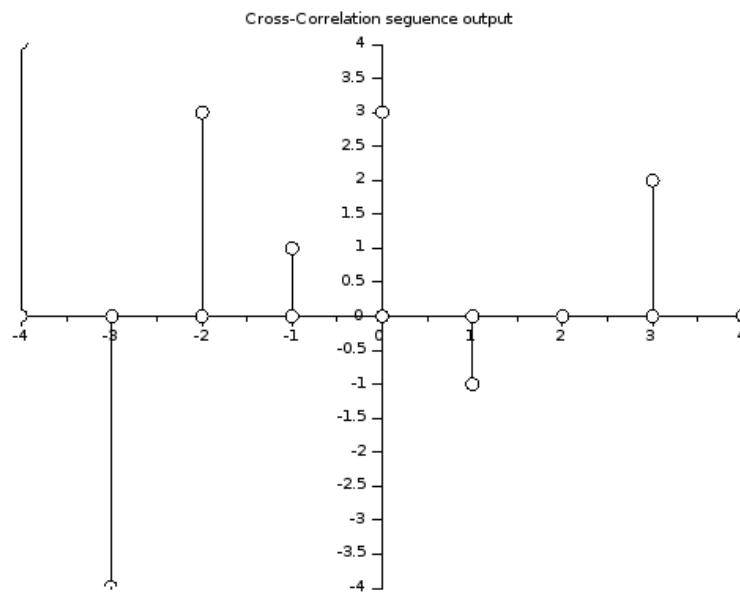


Figure 10.1: Cross Correlation

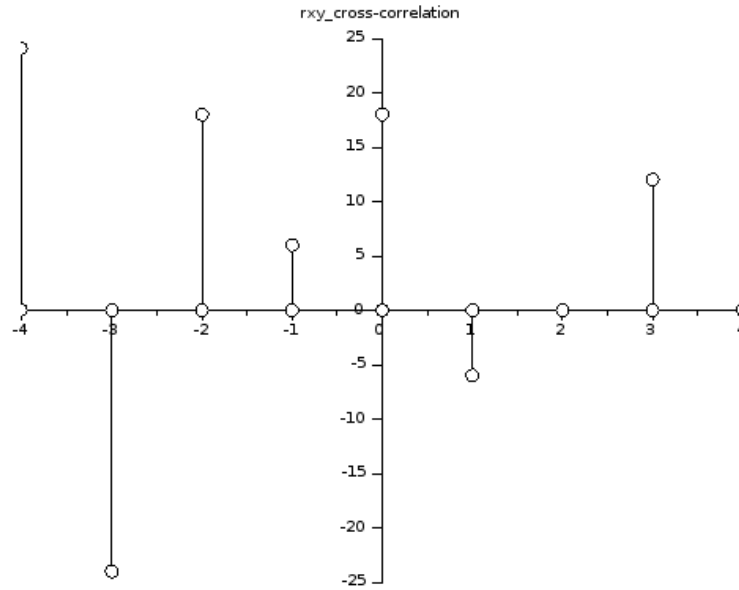


Figure 10.2: Cross Correlation Sequence

```
17 title('Cross-Correlation sequence output');
```

Scilab code Exa 10.2 Cross Correlation Sequence

```
1 //Scilab Code for Example 10.2 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //Cross Correlation
4 clear;
5 clc;
6 x=[4, -2, 2, 0, 4];
```

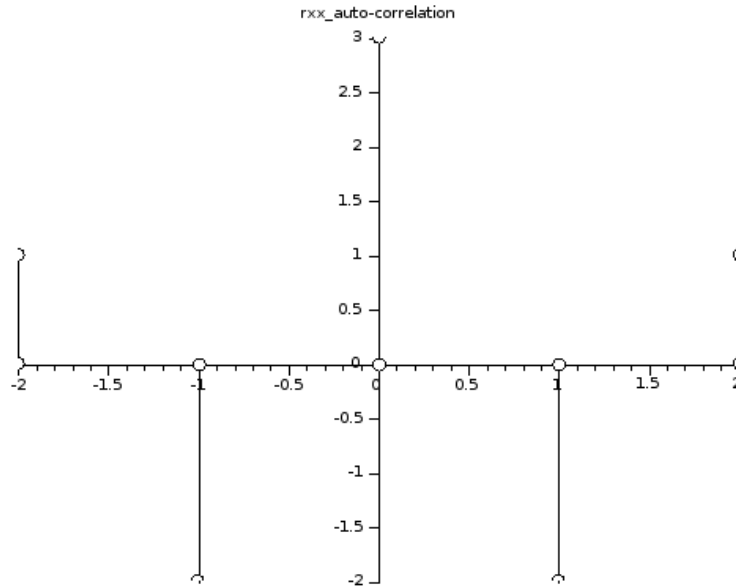


Figure 10.3: Auto Correlation

```

7 y=[0,3,0,-3,6];
8 //computation of cross correlation sequence;
9 n1 = max(size(y))-1;
10 n2 = max(size(x))-1;
11 r = xcorr(x,y,n1);
12 n=-4:4;
13 a=gca();
14 a.x_location="origin";
15 a.y_location="origin";
16 plot2d3(n,r,-9);
17 title('rxy_cross-correlation');

```

Scilab code Exa 10.4 Auto Correlation

```
1 //Scilab Code for Example 10.4 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Auto Correlation
4 clear;
5 clc;
6 x=[-1,1,-1];
7 //computation of auto correlation sequence;
8 r = xcorr(x);
9 n=-2:2;
10 a=gca();
11 a.x_location="origin";
12 a.y_location="origin";
13 plot2d3(n,r,-9);
14 title('rxx_auto-correlation');
```

Scilab code Exa 10.5 Auto Correlation

```
1 //Scilab Code for Example 10.5 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Auto Correlation
4 clear;
5 clc;
6 clear x n a;
7 k=1;
8 a=0.8;
9 for n=-30:30;
10     x(k)=a^(-n)*u(-n);
11     k=k+1;
12 end
```

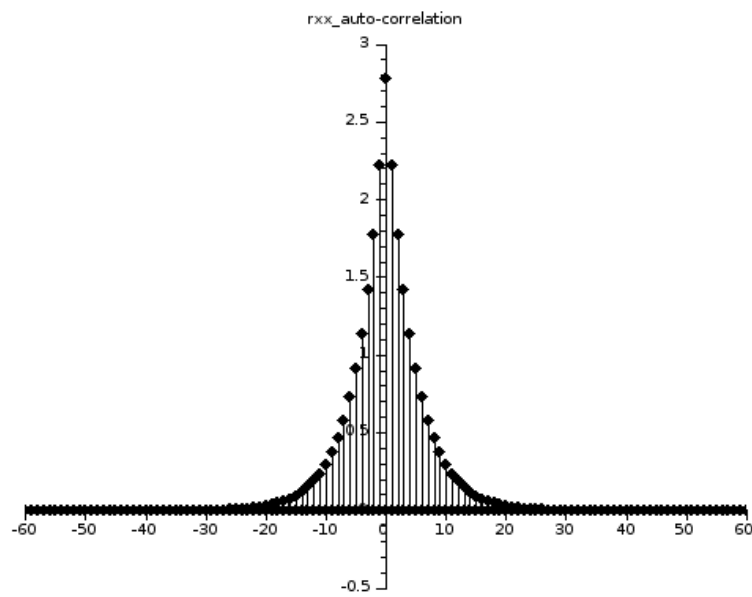


Figure 10.4: Auto Correlation

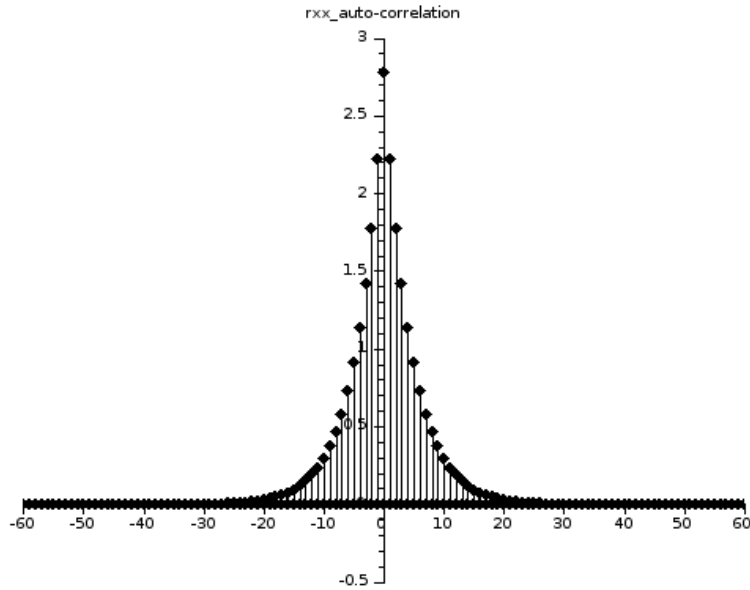


Figure 10.5: Auto Correlation

```

13 length(x)
14 //computation of auto correlation sequence;
15 r = xcorr(x);
16 n=-60:60;
17 a=gca();
18 a.x_location="origin";
19 a.y_location="origin";
20 plot2d3(n,r,-4);
21 title('rxx_auto-correlation');

```

Scilab code Exa 10.8 Auto Correlation

```

1 //Scilab Code for Example 10.8 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //Auto Correlation
4 clear;
5 clc;
6 clear x n a;
7 k=1;
8 a=0.8;
9 for n=-30:30;
10     x(k)=a^(n)*u(n);
11     k=k+1;
12 end
13 length(x)
14 //computation of auto correlation sequence;
15 r = xcorr(x);
16 n=-60:60;
17 a=gca();
18 a.x_location="origin";
19 a.y_location="origin";
20 plot2d3(n,r,-4);
21 title('rxx_auto-correlation');

```

Scilab code Exa 10.11 System response

```

1 //Scilab Code for Example 10.11 of Signals and
  systems by
2 //P.Ramakrishna Rao
3 //System Identification
4 //Cross Correlation
5 clear;
6 clc;
7 clear x n a y h z;

```

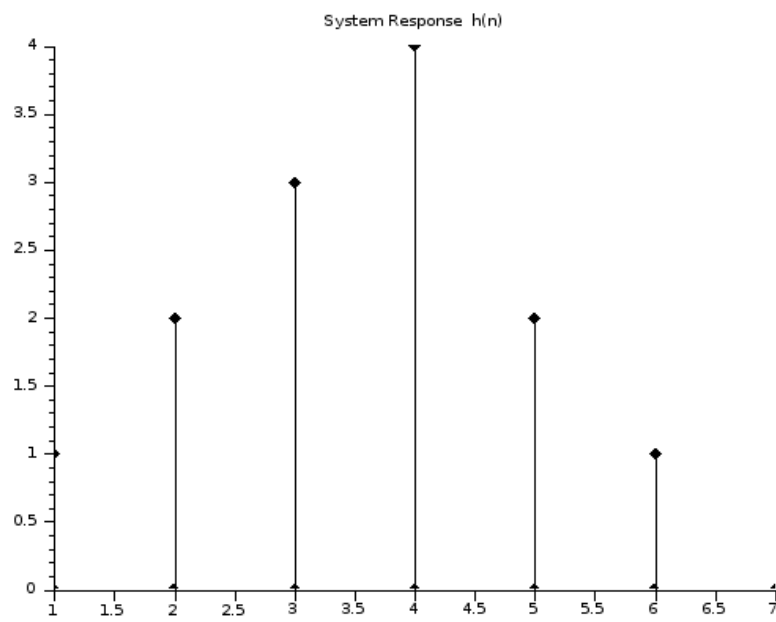


Figure 10.6: System response

```
8 x=[1,1,1,1,0,0];
9 y=[1,3,6,10,12,12,10,6,3,1];
10 h(1)=y(1)/x(1);
11 for n=2:7
12     for k=1:n-1
13         z(k)=(h(k)*x(n-k));
14     h(n)=(y(n)-sum(z));
15     end
16 end
17 disp(h, 'h(n) ');
18 n=1:7;
19 a=gca();
20 a.x_location="origin";
21 a.y_location="origin";
22 plot2d3(n,h,-4);
23 title('System Response h(n)');
```

Chapter 11

Hilbert Transform Continuous and Discrete

Scilab code Exa 11.1 Hilbert Transform

```
1 //Scilab Code for Example 11.1 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Hilbert Transform
4 clc;
5 clear;
6 n=1;
7 for t=-1:0.01:1
8     xr(n)=sin(2*%pi*t);
9     n=n+1;
10 end
11 //Computing Hilbert Transform
12 x=hilbert(xr);
13 t=-1:0.01:1;
14 plot(t,xr);
```

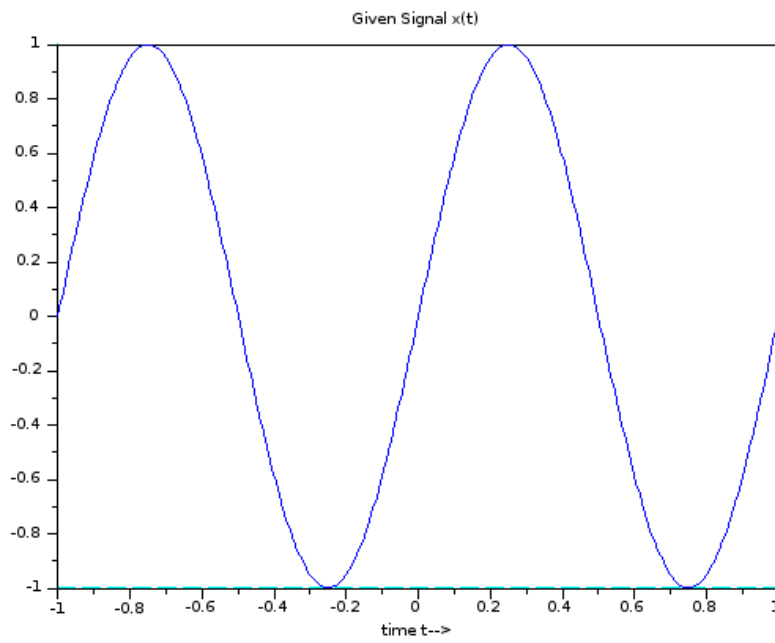


Figure 11.1: Hilbert Transform

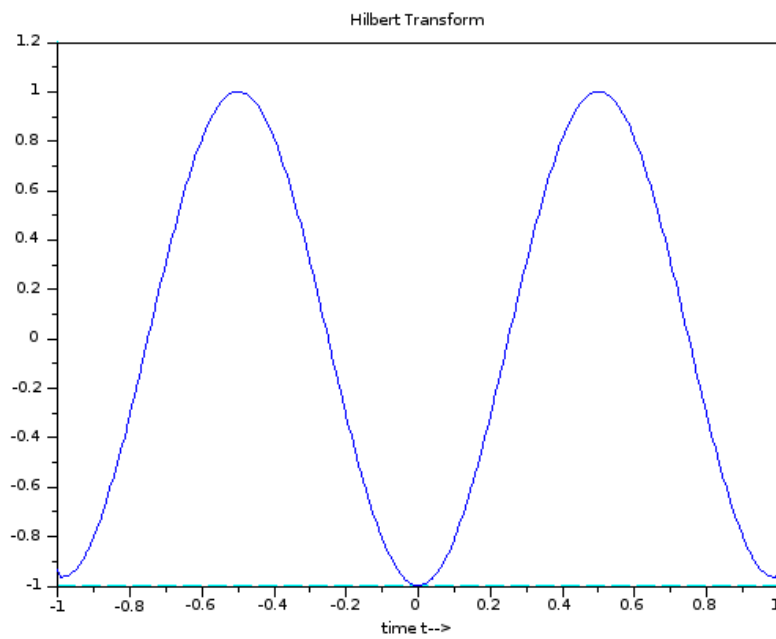


Figure 11.2: Hilbert Transform

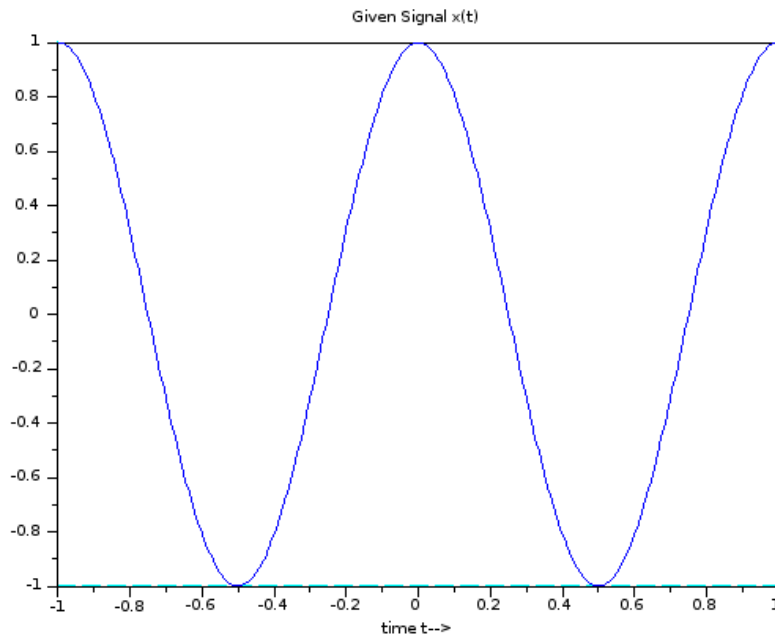


Figure 11.3: Hilbert Transform

```

15 title('Given Signal x(t)');
16 xlabel('time t-->');
17 figure(1);
18 t=-1:0.1:1;
19 plot(t, imag(x));
20 title('Hilbert Transform');
21 xlabel('time t-->');

```

Scilab code Exa 11.2 Hilbert Transform

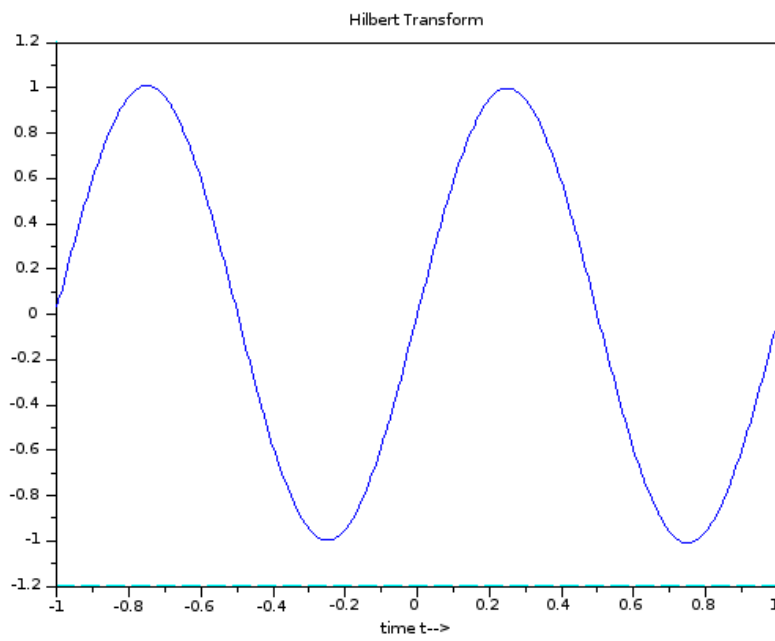


Figure 11.4: Hilbert Transform

```

1 //Scilab Code for Example 11.2 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Hilbert Transform
4 clc;
5 clear;
6 n=1;
7 for t=-1:0.01:1
8     xr(n)=cos(2*%pi*t);
9     n=n+1;
10 end
11 //Computing Hilbert Transform
12 x=hilbert(xr);
13 t=-1:0.01:1;
14 plot(t,xr);
15 title('Given Signal x(t)');
16 xlabel('time t—>');
17 figure(1);
18 t=-1:0.01:1;
19 plot(t,imag(x));
20 title('Hilbert Transform');
21 xlabel('time t—>');

```

Scilab code Exa 11.3 Hilbert Transform

```

1 //Scilab Code for Example 11.3 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Hilbert Transform
4 clc;
5 clear xr n t x1 x2;

```

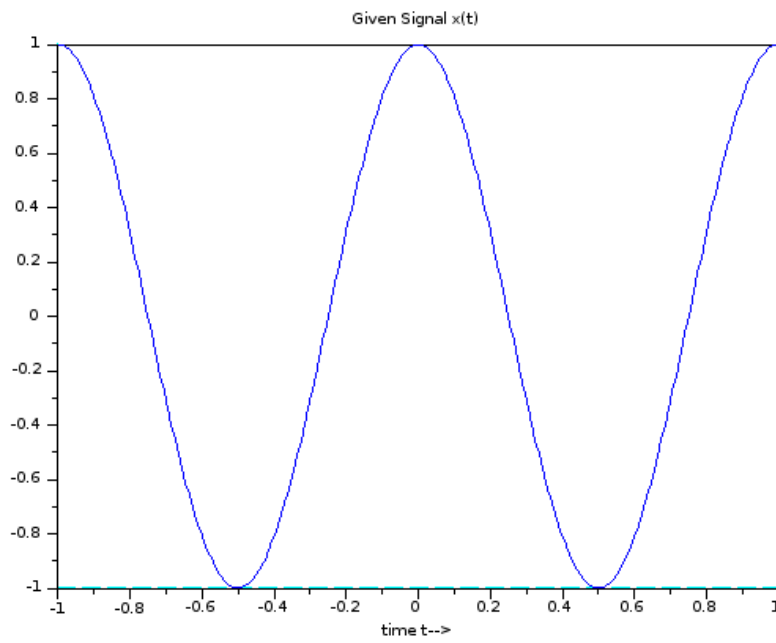


Figure 11.5: Hilbert Transform

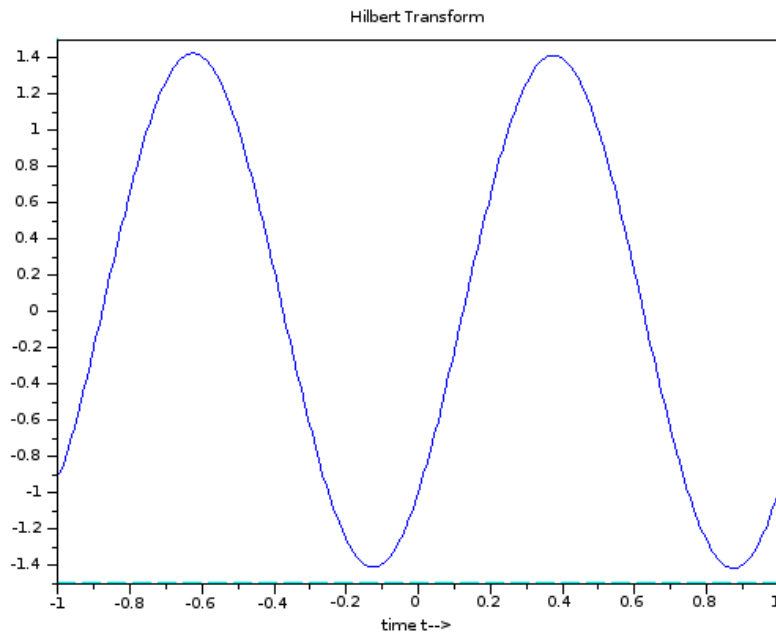


Figure 11.6: Hilbert Transform

```

6 clear;
7 n=1;
8 for t=-1:0.01:1
9     xr(n)=exp(%i*2*%pi*t);
10    n=n+1;
11 end
12 //Computing Hilbert Transform
13 x1=hilbert(real(xr));
14 x2=hilbert(imag(xr));
15 x=x1+x2;
16 t=-1:0.01:1;
17 plot(t,xr);
18 title('Given Signal x(t)');
19 xlabel('time t-->');
20 figure(1);
21 t=-1:0.01:1;
22 plot(t,imag(x));
23 title('Hilbert Transform');
24 xlabel('time t-->');

```

Scilab code Exa 11.6 Hilbert Transform

```

1 //Scilab Code for Example 11.6 of Signals and
   systems by
2 //P.Ramakrishna Rao
3 //Hilbert Transform
4 clc;
5 clear A T t x;
6 T=2;
7 A=1;
8 n=1;

```

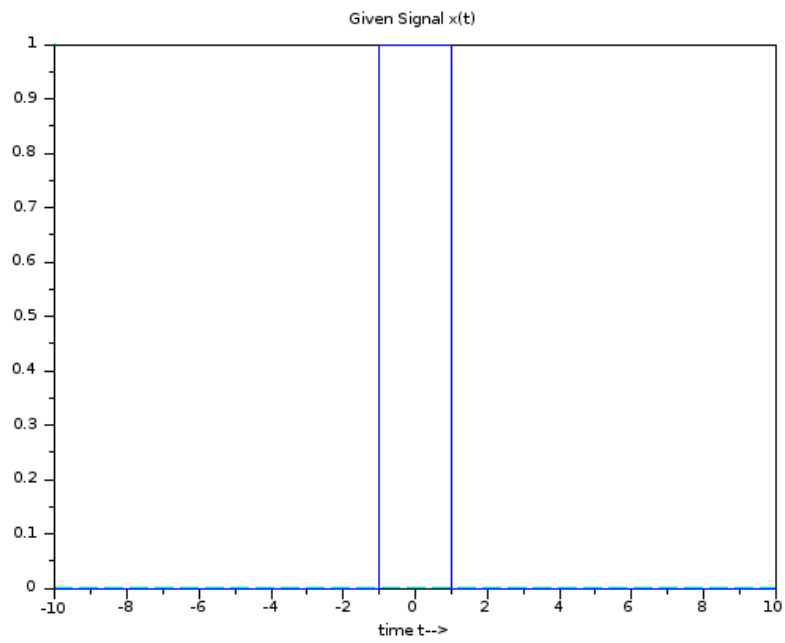


Figure 11.7: Hilbert Transform

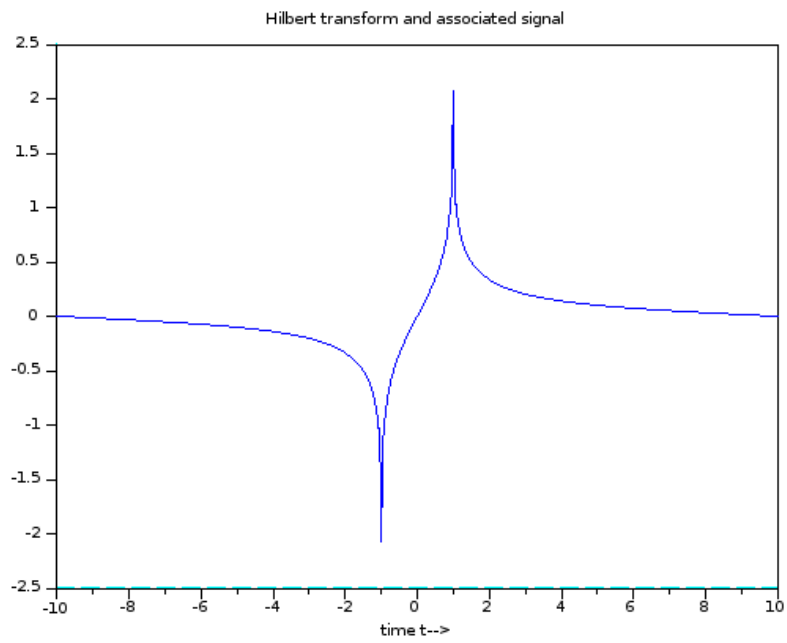


Figure 11.8: Hilbert Transform

```

9  for t=-10:0.01:10;
10     if t>-T/2 & t<T/2 then
11         x(n)=A;
12         n=n+1;
13     else x(n)=0;
14         n=n+1;
15     end
16 end
17 t=-10:0.01:10;
18 //Signal x(t)
19 plot(t,x);
20 title('Given Signal x(t)');
21 xlabel('time t—>');
22 //Computing Hilbert Transform
23 xr=hilbert(x);
24 xr_imag=imag(xr);
25 figure(1);
26 t=-10:0.01:10;
27 plot(t,xr_imag);
28 title('Hilbert transform and associated signal');
29 xlabel('time t—>');

```
