Scilab Textbook Companion for Signals And Systems by P. R. Rao[1]

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

May 24, 2016

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

Book Description

Title: Signals And Systems

Author: 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

List of Figures

Chapter 1

Signals Continuous and Discrete time

Scilab code Exa 1.1 Finding Fundamental Period

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

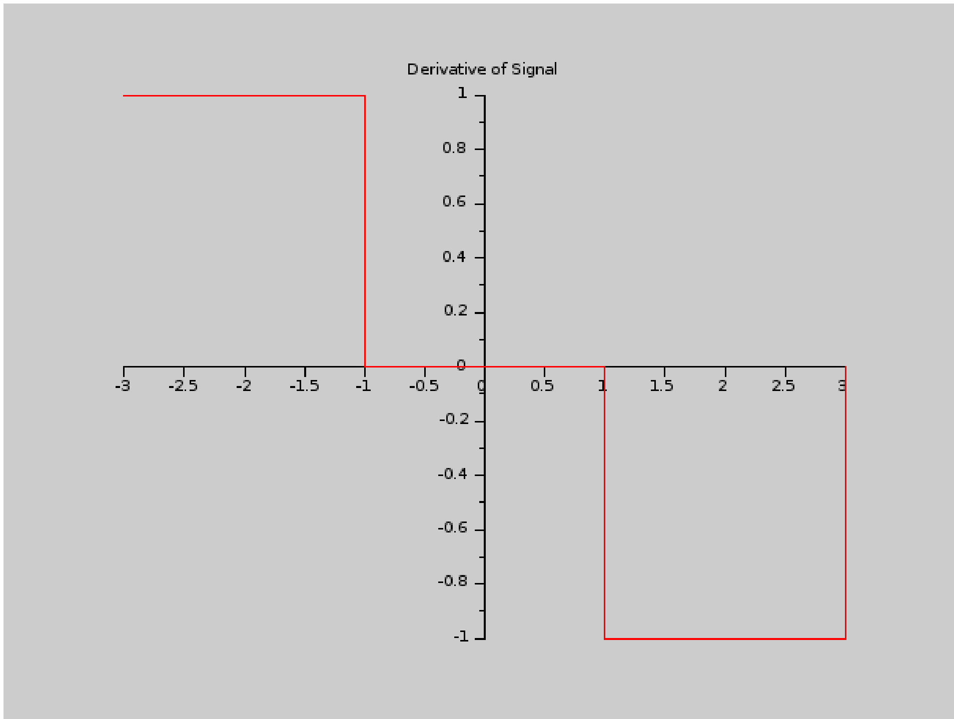

Figure 1.1: Finding Fundamental Period

Scilab code Exa 1.3.a Periodicity

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

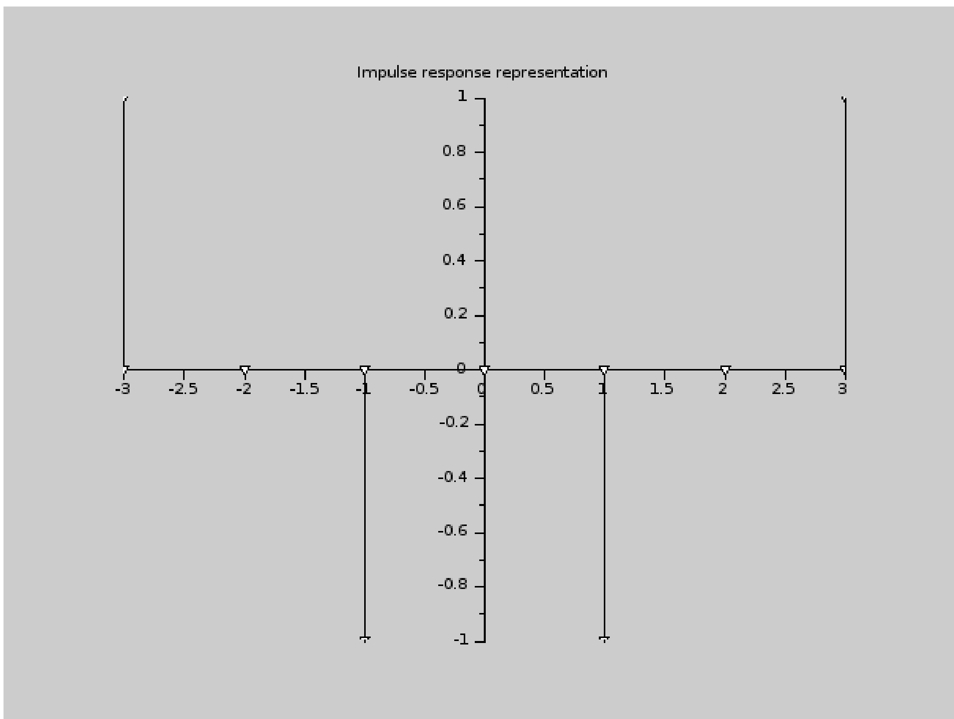

Figure 1.2: Finding Fundamental Period

```
clc;
clear;
syms t;
x=3*cos(0.4*%pi*t)+2*sin(0.66*t);
disp(x,'x(t)');
t=0:1/50:50;
x=3*cos(0.4*%pi*t)+2*sin(0.66*t);
t=0:1/50:50;
plot(t,x);
title('x(t)');
xlabel('Time in seconds');
disp('plotted the signal and shown that it is not periodic and is increasing');
```

Scilab code Exa 1.3.b Periodicity and fundamental period

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

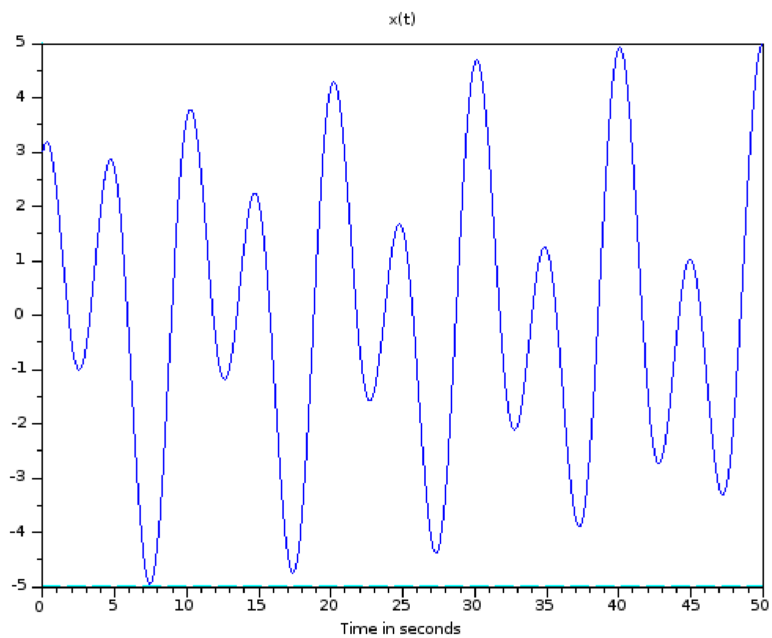

Figure 1.3: Periodicity

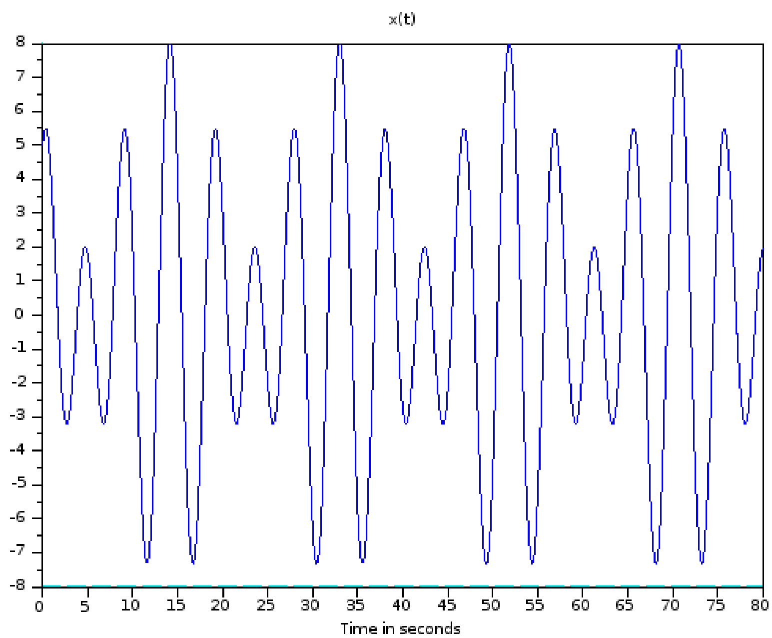

Figure 1.4: Periodicity and fundamental period

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

Scilab code Exa 1.4 Fundamental Period

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

```
//x(t)=cos(7*t)+sin(4*t)
clc;
clear;
syms t;
x=sin(4*t)+cos(7*t);
disp(x,'x(t)');
t=0:1/12:12
x=sin(4*t)+cos(7*t);
t=0:1/12:12;
plot(t,x);
title('x(t)');
xlabel('Time in seconds');
disp('plotted the signal and shown that it is
    periodic with period of 2pi');
```

Scilab code Exa 1.5 Fundamental Period

```
//Scilab Code for Example 1.5 of Signals and systems
    by
//P.Ramakrishna Rao
//Determine whether the given signal is periodic or
    not
//x(t)=cos(t)+sin(sqrt(2)*t)
clc;
clear;
syms t;
x=cos(t)+sin(sqrt(2)*t);
disp(x,'x(t)');
for t=0:1:100;
x(t+1)=cos(t)+sin(sqrt(2)*t);
end
t=0:1:100;
plot(t,x);
```

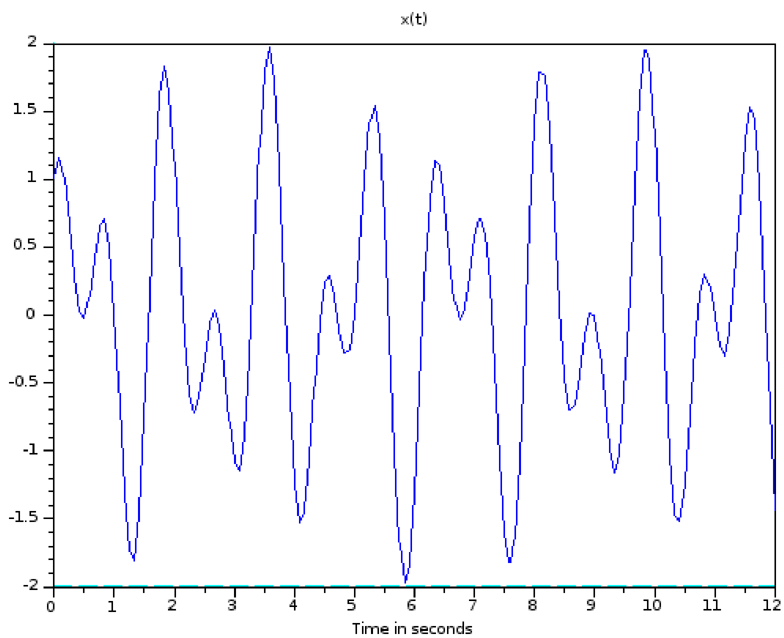

Figure 1.5: Fundamental Period

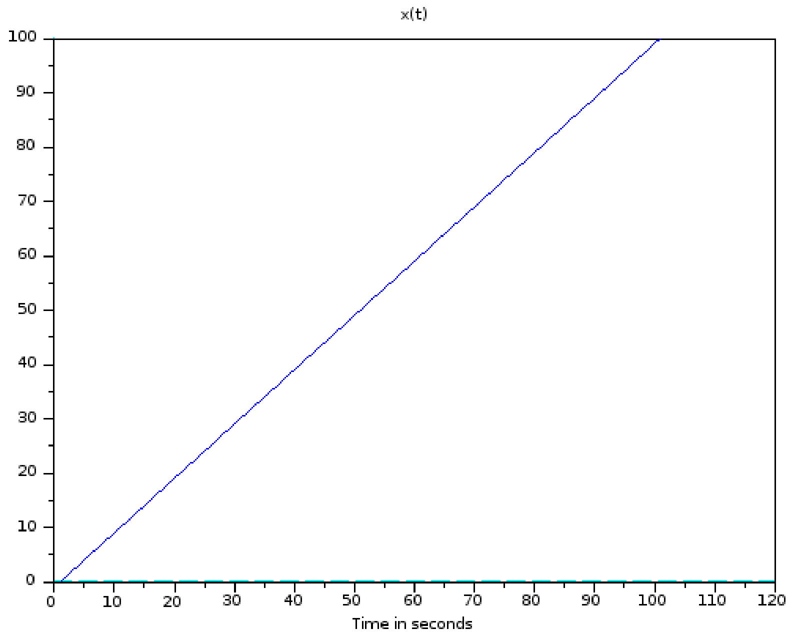

Figure 1.6: Fundamental Period

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

Scilab code Exa 1.6 Fundamental Period

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

```
not
clc;
clear;

n=0:1:10;
x(n+1)=2*sin(0.8*%pi*n);
a=gca();
a.x_location="origin";
a.y_location="origin";
n=0:1:10;
plot2d3(n,x,9);
title('x(n)');
disp('ploting the signal and showing that it is periodic with period of 5');
```

Scilab code Exa 1.7 Even and odd Components

```
//Scilab Code for Example 1.7 of Signals and systems by
//P.Ramakrishna Rao
clear;
clc;
n=1;
for t=-10:0.1:10;
    //Function for Even signal
    y1(n)=0.5*(exp(-t)*u(t)+exp(t)*u(-t));
    n=n+1;
end
a=gca();
a.x_location="origin";
a.y_location="origin";
t=-10:0.1:10;
//Plot of Even Signal
```

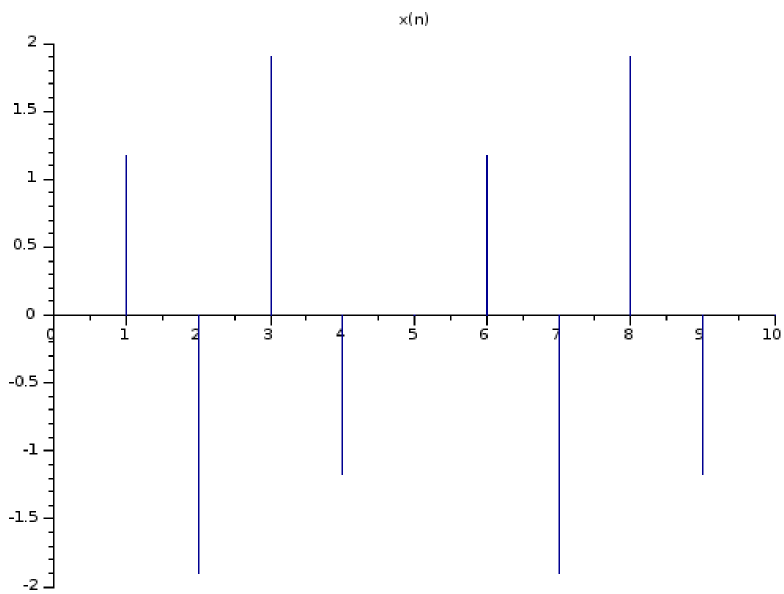

Figure 1.7: Fundamental Period

```
plot(t,y1);
title('y1(t)');
xlabel('Time in seconds');
n=1;
for t=-1:0.01:1;
    //Function for Odd signal
    y2(n)=0.5*(exp(-t)*u(t)-exp(t)*u(-t));
    n=n+1;
end
figure(1);
a=gca();
a.x_location="origin";
a.y_location="origin";
t=-1:0.01:1;
//Plot of Odd Signal
plot(t,y2)
disp('plotted the signal both in even and odd forms');
title('y2(t)');
xlabel('Time in seconds');
```

Scilab code Exa 1.11 Waveforms

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

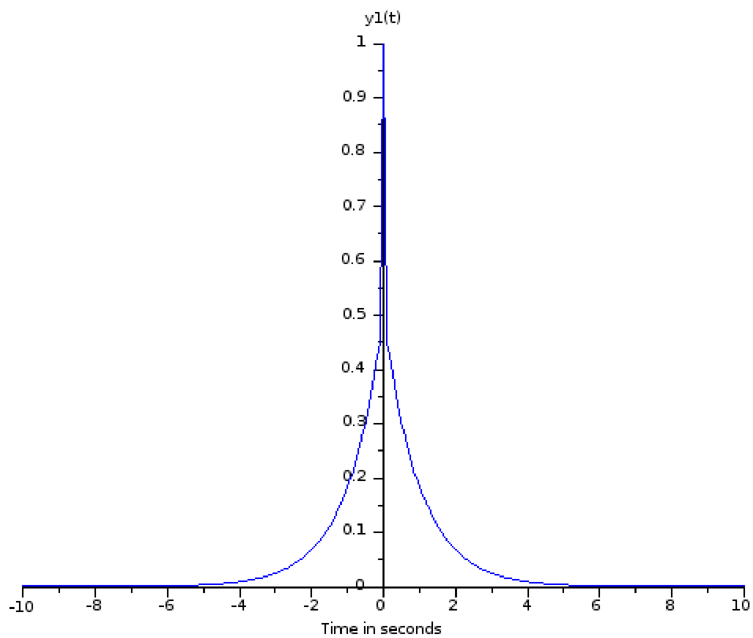

Figure 1.8: Even and odd Components

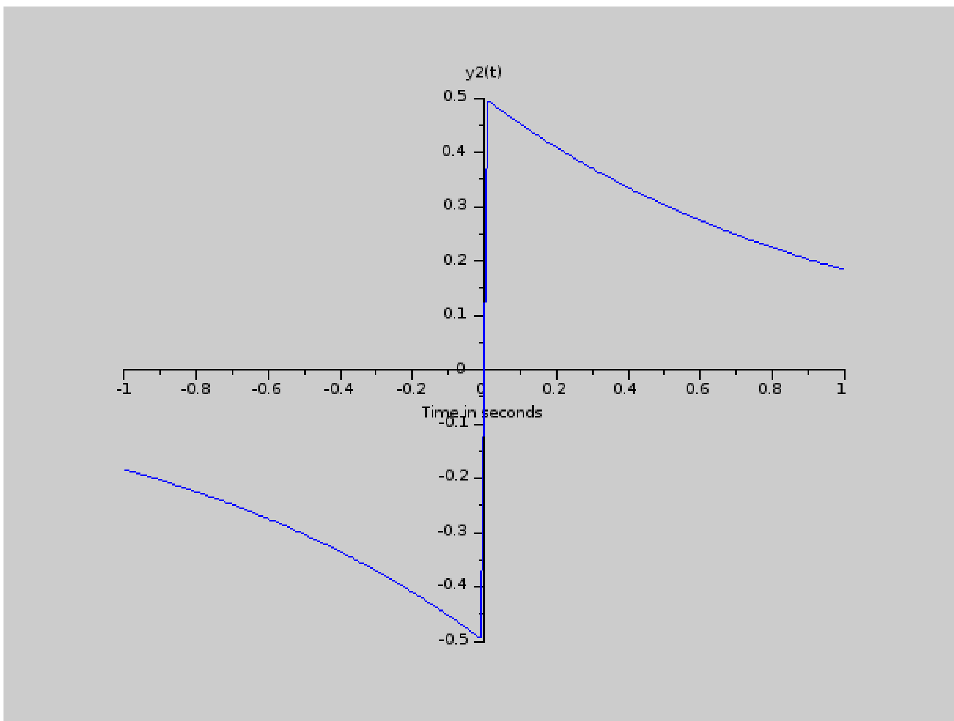

Figure 1.9: Even and odd Components

```
    -1)+abs(t-3)*u(t-3);
end
t=-3:1:3;
//for the main given signal
a=gca();
a.x_location="origin";
a.y_location="origin";
plot(t,y);
title('Main Signal)');
dy=0*y;
for i=1:6
dy(i)=(y(i+1)-y(i))/1;
end
//for the derivative of the given signal
figure(1);
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d2(t,dy);
title('Derivative of Signal');
dy2=0*dy;
dy2(1)=dy(1)-0;
for i=1:6
dy2(i+1)=(dy(i+1)-dy(i))/1;
end
//for the impulse response representation or second
    derivative
figure(2);
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d3(t,dy2,-5);
title('Impulse response representation');
```

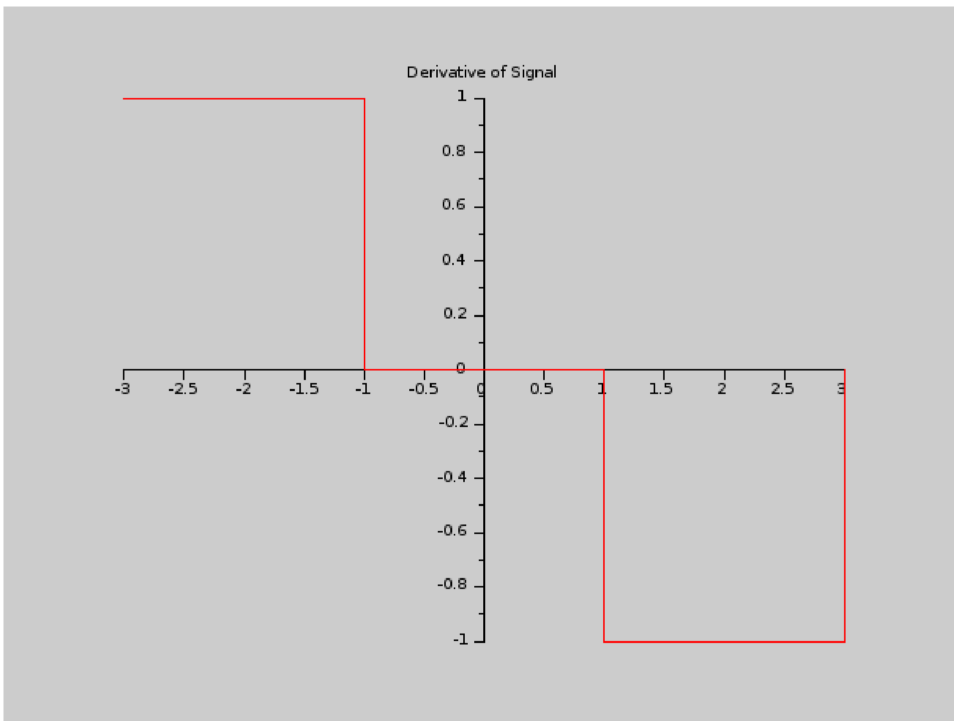

Figure 1.10: Waveforms

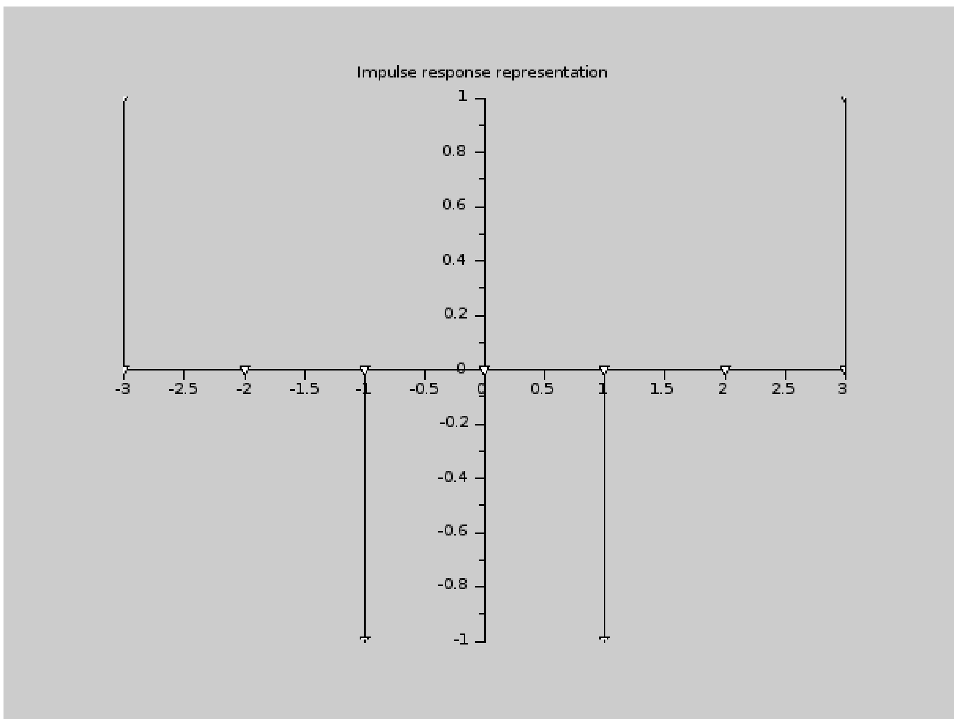

Figure 1.11: Waveforms

Scilab code Exa 1.12 Time Scaling

```
//Scilab Code for Example 1.12 of Signals and
   systems by
//P.Ramakrishna Rao
clc;
clear;
x=-2:1:3;
y=[-1.5,2,2,1,-1.5,2.5];
//Plot of x(n)
plot2d3(x,y,5);
xtitle ( 'Time Scaling x(n)');
a = gca(); // get the current axes
a.x_location = "origin";
a.y_location = "origin";
x=-2:1:3;
y=[0,-1.5,2,-1.5,0,0];
figure(1);
//Plot of x(2n)
plot2d3(x,y,5);
a = gca(); // get the current axes
a.x_location = "origin";
a.y_location = "origin";
xtitle ( 'Time Scaling x(2n)' );
```

Scilab code Exa 1.13 Plot at a particular sampling frequency

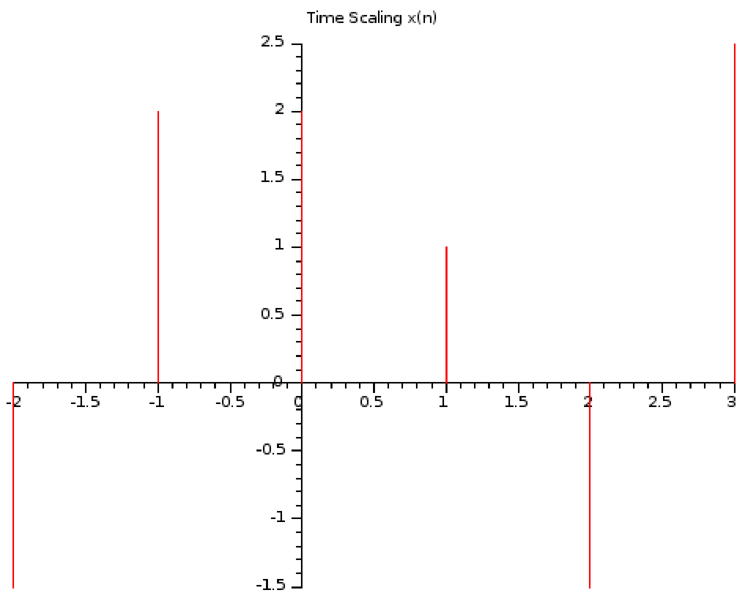

Figure 1.12: Time Scaling

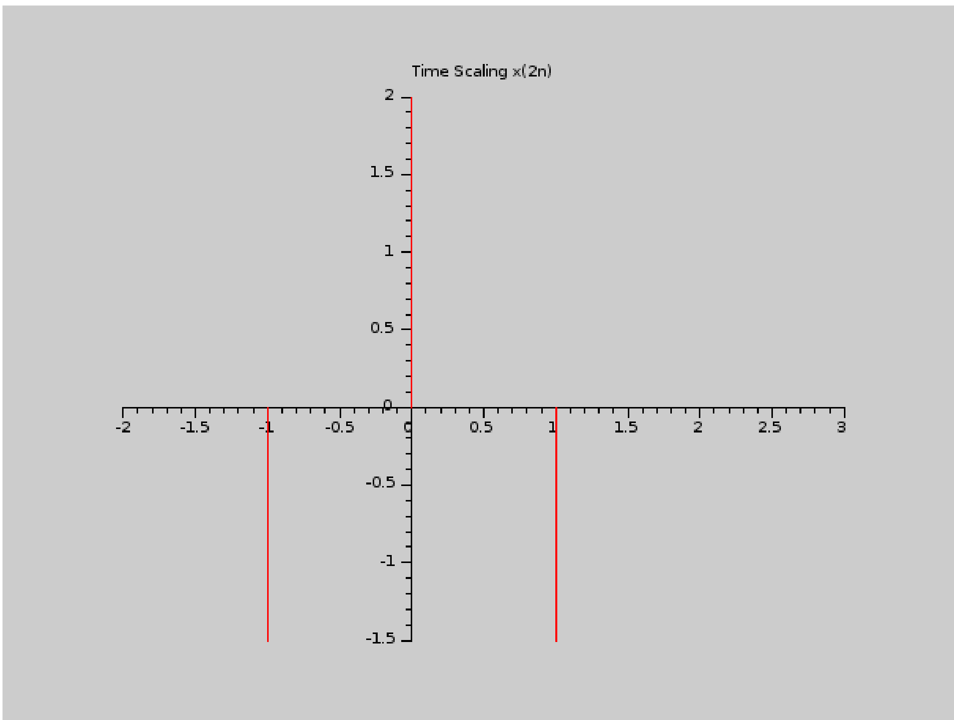

Figure 1.13: Time Scaling

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

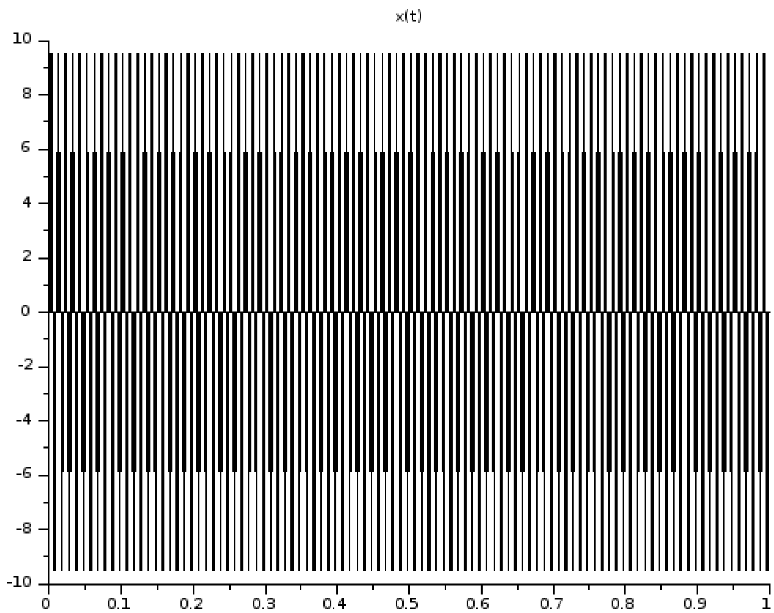

Figure 1.14: Plot at a particular sampling frequency

Chapter 2

Laplace and Z Transform

Scilab code Exa 2.1 Laplace Transform

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

Scilab code Exa 2.2 Inverse Laplace

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

```
//P.Ramakrishna Rao
//Inverse Laplace Transform
clc;
clear;
syms t s
X=(s+2)/(s^2+8*s+25)
f4=ilaplace(X);
disp(f4*'u(t)',"x(t)=");
//t>=0
```

Scilab code Exa 2.3 Initial Value Theorem

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

Scilab code Exa 2.4 Laplace Transform

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

Scilab code Exa 2.6 Circuit Current

```
//Scilab Code for Example 2.6 of Signals and systems by
//P.Ramakrishna Raoclear;
clc;
clear;
syms s X x t R C V Vo;
//After solving for I(s)
//I(s)=(V-Vo)/R . 1/(s+1/RC)
X=(V-Vo)/((s+1/(R*C))*R);
disp(X,"I(s)=");
x=ilaplace(X);
disp(x,"i(t)=");
```

Scilab code Exa 2.7 Inverse Laplace Transform

```
//Scilab Code for Example 2.7 of Signals and systems by
//P.Ramakrishna Rao
//Unilateral Laplace Transform using partial fraction
clear;
clc;
syms t s;
s= %s;
a1=pfss((s+3)/(s^2+3*s+2))
f1=ilaplace(a1(1))
f2=ilaplace(a1(2))
fy=f1+f2
disp(fy*'u(t)',"i) f(t)=")
a2=pfss((2*s-1)/(s^2+2*s+1))
a2(1)=2/(s+1)
a2(2)=-3/(s+1)^2
f1=ilaplace(a2(1))
f2=ilaplace(a2(2))
fz=f1+f2
disp(fz*'u(t)',"ii) f(t)=")
```

Scilab code Exa 2.8 Inverse Laplace Transform

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

Scilab code Exa 2.9 Circuit Current

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

Scilab code Exa 2.10 Laplace Transform

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

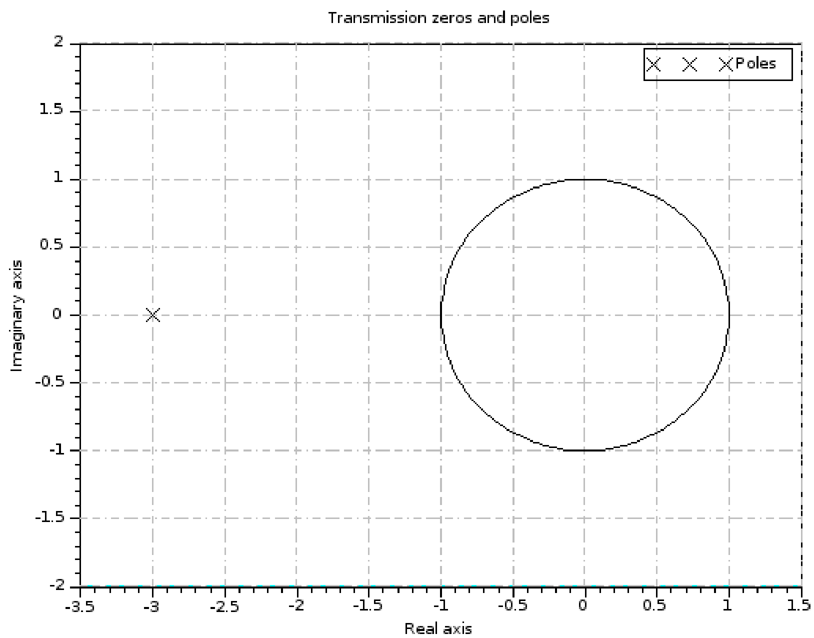

Figure 2.1: Laplace Transform

Scilab code Exa 2.15 Laplace Transform

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

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

Scilab code Exa 2.16 Laplace Transform

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

Scilab code Exa 2.17 Laplace Transform

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

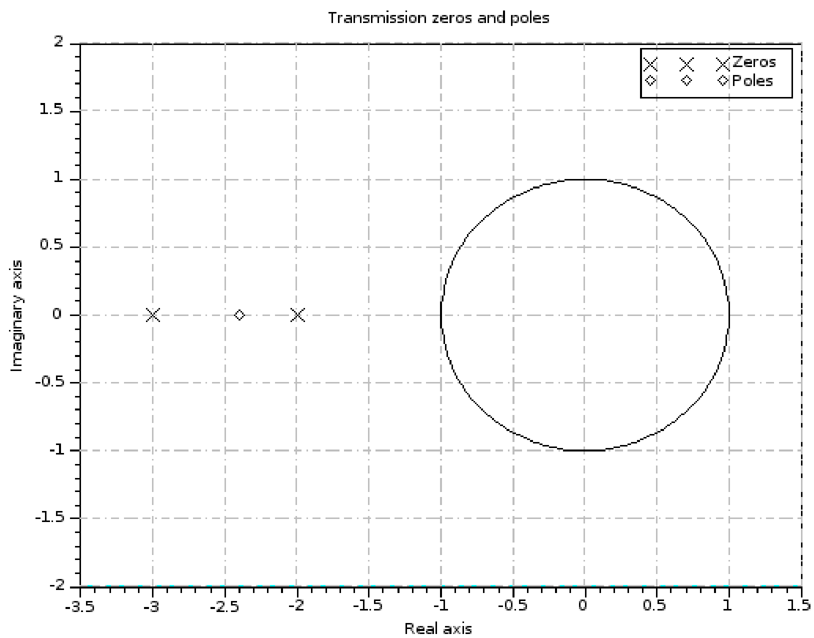

Figure 2.2: Laplace Transform

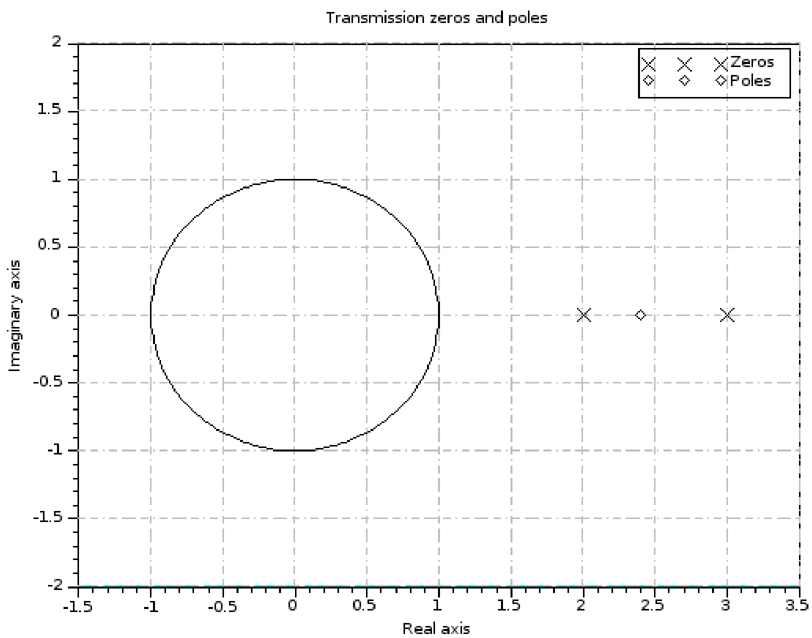

Figure 2.3: Laplace Transform

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

Scilab code Exa 2.19 Laplace Transform

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

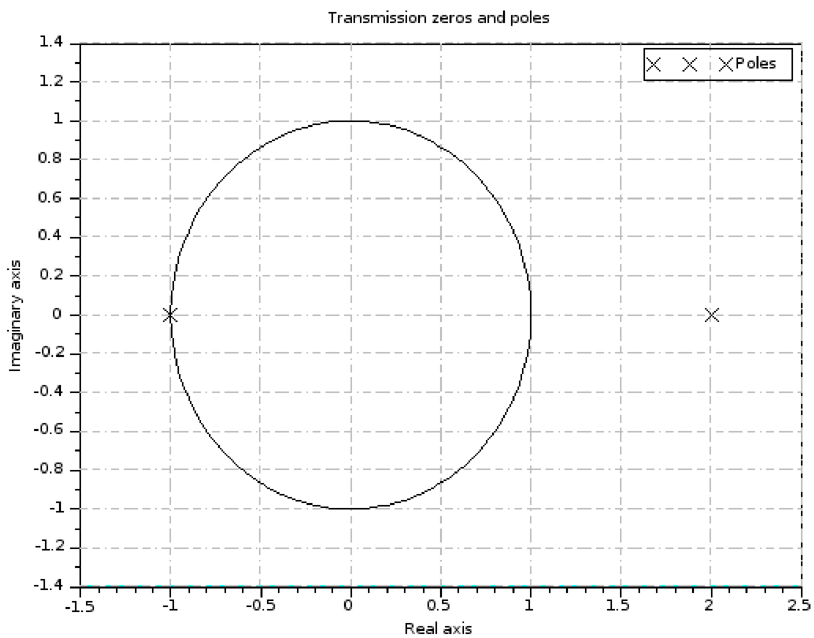

Figure 2.4: Laplace Transform

```
s=%s;
y=laplace(exp(-t)-exp(2*t),t,s);
disp(y,"X(s)=");
y=(1/(s+1))-(1/(s-2));
plzr(y);
```

Scilab code Exa 2.21 Unilateral Z Transform

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

```
//Z- transform of a^n u(n)
clear;
clc ;
close ;
syms a n z;
x1 =1/2;
x2=1/3;
X1= symsum (x1*(z^(-n)),n ,0, %inf );
X2= symsum (x2*(z^(-n)),n ,0, %inf );
X=X1+X2;
//Display the result
disp (X,"Z-transform of u(n) is:");
disp('ROC is the Region |z|> 1/2');
```

Scilab code Exa 2.29 Z Transform

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

Scilab code Exa 2.37 Inverse Z Transform

```
//Scilab Code for Example 2.37 of Signals and
    systems by
//P.Ramakrishna Rao
//Inverse Z Transform:ROC 1<|z|<2
clear;
clc;
z = %z;
syms n z1;
//To find out Inverse z transform z must be linear z
     = z1
X  =(z*(z^2-z+1))/((z-0.5)*(z-2)*(z-1))
X1 = denom(X);
zp = roots(X1);
X1 = (z1*(z1^2-z1+1))/((z1-0.5)*(z1-2)*(z1-1))
F1 = X1*(z1^(n-1))*(z1-zp(1))
F2 = X1*(z1^(n-1))*(z1-zp(2))
F3 = X1*(z1^(n-1))*(z1-zp(3))
h1 = limit(F1,z1,zp(1));
disp(h1*'u(-n-1)','h1[n]=')
h2 = limit(F2,z1,zp(2));
disp((h2)*'u(n)','h2[n]=')
h3 = limit(F3,z1,zp(3));
disp((h3)*'u(n)','h3[n]=')
disp((h3)*'u(n)'+(h2)*'u(n)'-(h1)*'u(-n-1)','h[n]=')
////Result
//  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

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

```
clc;
z = %z;
syms n z1;
//To find out Inverse z transform z must be linear z
    = z1
X  =z^2/(z^2+3*z+2);
X1 = denom(X);
zp = roots(X1)
X1 = z1^2/(z1^2+3*z1+2);
F1 = X1*(z1^(n-1))*(z1-zp(1));
F2 = X1*(z1^(n-1))*(z1-zp(2));
h1 = limit(F1,z1,zp(1));
disp(h1*'u(n)','h1[n]=')
h2 = limit(F2,z1,zp(2));
disp((h2)*'u(n)','h2[n]=');
disp((h1)*'u(n)'+(h2)*'u(n)','h[n]=');
////Result
//  h[n]= (2(- 2)^n+1 - (- 1)^n)*u(n)
```

Scilab code Exa 2.38.b Inverse Z Transform

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

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

Scilab code Exa 2.39 Inverse Z Transform

```
//Scilab Code for Example 2.39 of Signals and
  systems by
//P.Ramakrishna Rao
//Inverse Z Transform:ROC |z|>2
clear;
clc;
z = %z;
syms n z1;
//To find out Inverse z transform z must be linear z
  = z1
X  =2/(z^-1+2);
X1 = denom(X);
zp = roots(X1)
X1 =z1/(z1+0.5);
F1 = X1*(z1^(n-1))*(z1-zp(1));
h1 = limit(F1,z1,zp(1))
disp(-(h1)*'u(-n-1)','h[n]=');
////Result
//  h[n]= (- (- 0.5)^n)*u(-n-1)
```

Scilab code Exa 2.40 Inverse Z Transform

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

```
//Inverse Z Transform:ROC |z|>2
clear;
clc;
z = %z;
syms n z1;
//To find out Inverse z transform z must be linear z = z1
X  =z^2/((z-0.5)*(z-1));
X1 = denom(X);
zp = roots(X1)
X1 = z1^2/((z1-0.5)*(z1-1));
F1 = X1*(z1^(n-1))*(z1-zp(1));
F2 = X1*(z1^(n-1))*(z1-zp(2));
h1 = limit(F1,z1,zp(1));
disp(h1*'u(n)','h1[n]=')
h2 = limit(F2,z1,zp(2));
disp((h2)*'u(n)','h2[n]=');
disp(-(h1)*'u(-n-1)'-(h2)*'u(-n-1)','h[n]=');
////Result
//  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

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

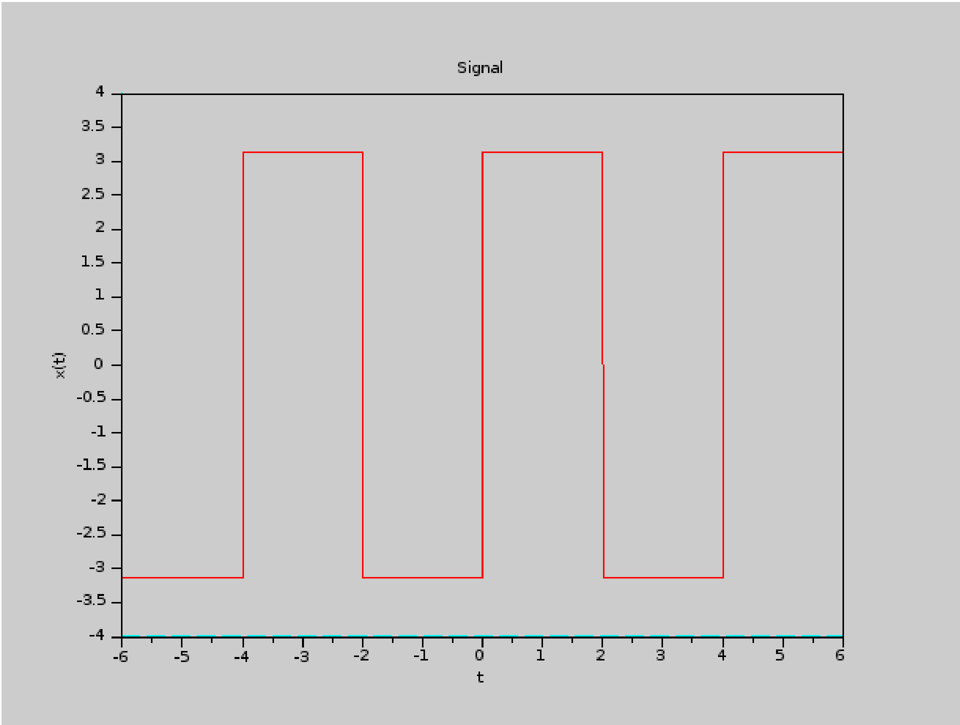

Figure 3.1: Complex Fourier Exponential Series

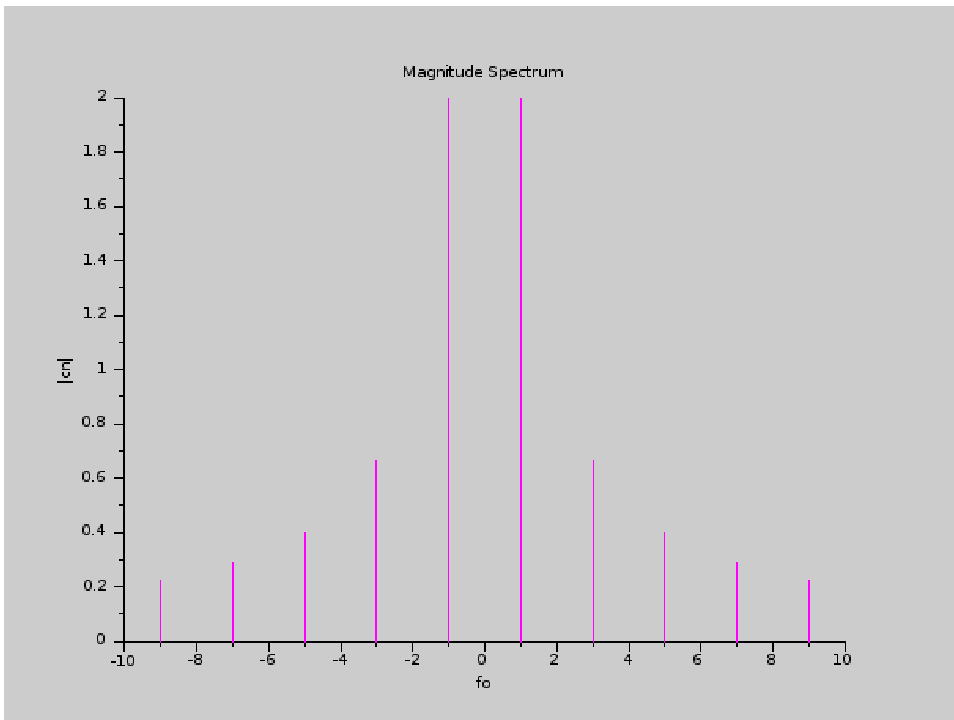

Figure 3.2: Complex Fourier Exponential Series

```
        x=[x -ones(1,length(t_temp))*%pi];
    else
        x=[x ones(1,length(t_temp))*%pi];
    end
end
figure(1);
title('Signal');
ylabel('x(t)');
xlabel('t');
plot(t,x,'r')
w0=%pi/2;
for k=-10:10
    cc(k+11,:)=exp(-%i*k*w0*t);
    ck(k+11)=x*cc(k+11,:)'/length(t);
    if abs(ck(k+11))<0.01 then
    ck(k+11)=0;
    else if real(ck(k+11))<0.1  then
    ck(k+11)=%i*imag(ck(k+11));
    end
    end
    if k==0 then
        c0=ck(k+11);
    end
end
q=abs(ck);
figure(2);
title('Magnitude Spectrum');
ylabel('|cn|');
xlabel('fo');
f=-10:1:10;
plot2d3(f,q)
```

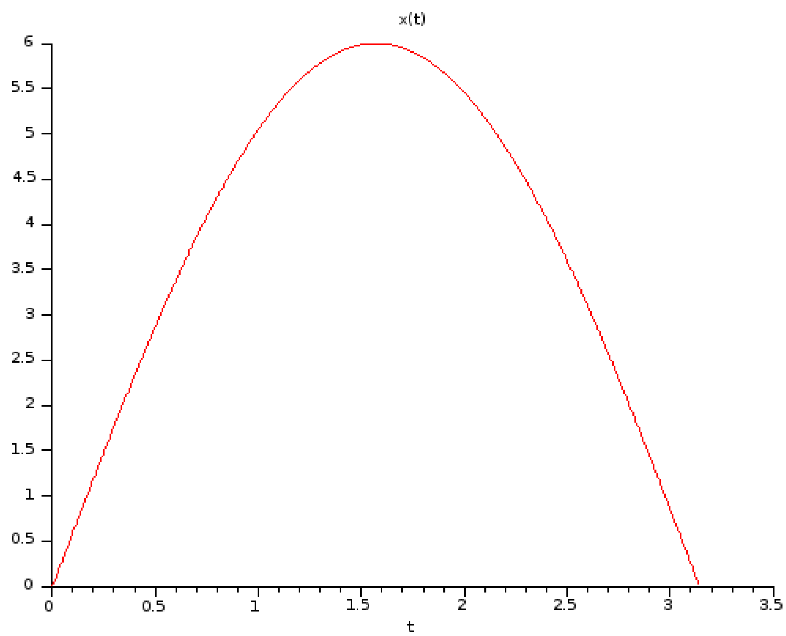

Figure 3.3: Complex Fourier Exponential Series

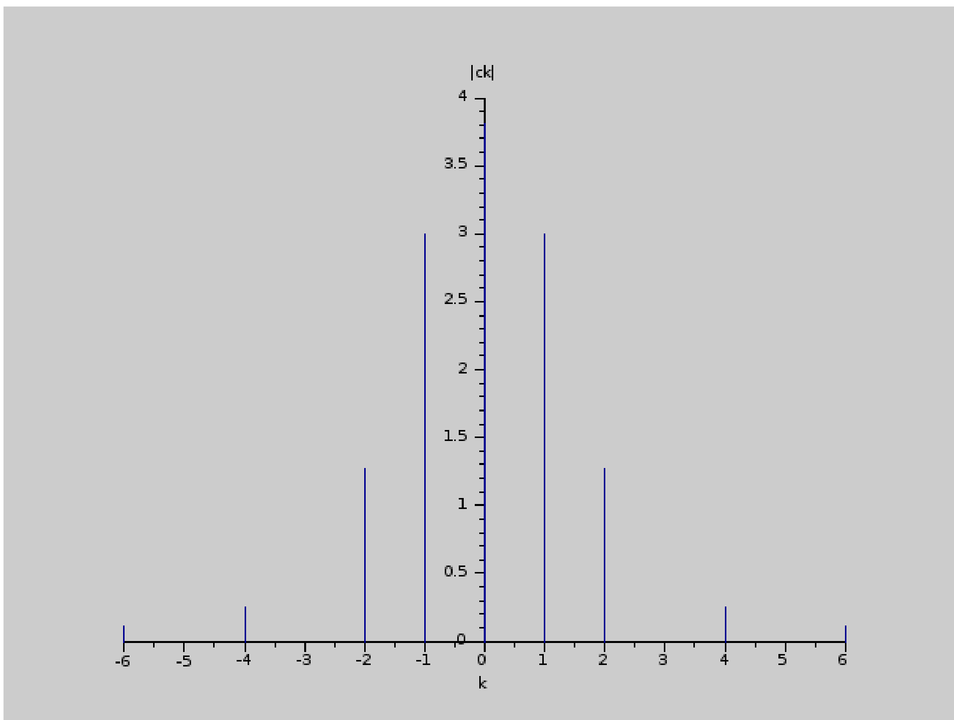

Figure 3.4: Complex Fourier Exponential Series

Scilab code Exa 3.14 Complex Fourier Exponential Series

```
//Scilab Code for Example 3.14 of Signals and
    systems by
//P.Ramakrishna Rao
//CTFS coefficients of a periodic signal
//x(wt) = Asin wt, 0<wt<pi, and 0, pi<|t|<2*pi
clear;
clc;
A=6;
T =2*%pi;
T1 = T/2;
t = 0:0.01:T1;
Wo = 2*%pi/T;
xt =A*sin(Wo*t);
for k =0:6
  C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
  c(k+1) = xt*C(k+1,:)'/length(t);
  if(abs(c(k+1))<=0.01)
    c(k+1)=0;
  end
end
c =c';
c_conj = real(c(:))-sqrt(-1)*imag(c(:));
ck = [c_conj($:-1:1)',c(2:$)];
k = 0:6;
k = [-k($:-1:1),k(2:$)];
figure(1);
c = gca();
c.y_location = "origin";
c.x_location = "origin";
//c.data_bounds=[-2,0;2,2];
plot2d(t,xt,5)
poly1 = c.children(1).children(1);
poly1.thickness = 3;
title('x(t)')
xlabel('t')
figure(2);
```

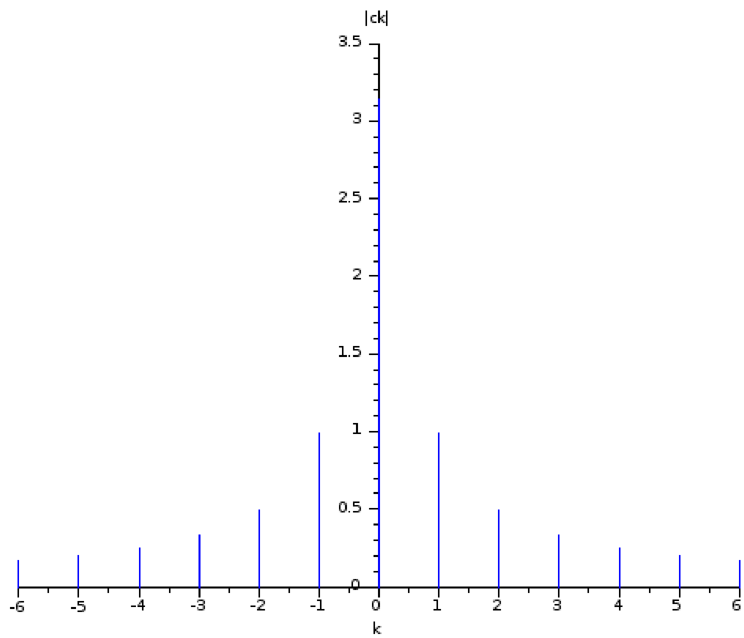

Figure 3.5: Complex Fourier Exponential Series

```
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot2d3('gnn',k,abs(ck),9)
poly1 = c.children(1).children(1);
poly1.thickness = 3;
title('|ck|')
xlabel('k')
```

Scilab code Exa 3.16 Complex Fourier Exponential Series

```
//Scilab Code for Example 3.16 of Signals and systems by
//P.Ramakrishna Rao
//A=%pi or 3.14
clear;
clc;
//Trignometric Fourier Coefficients
for n=0:5
a(n+1)=integrate('t*cos(2*%pi*n*t)','t',0,1);
end
for n=0:5
b(n+1)=integrate('t*sin(2*%pi*n*t)','t',0,1);
end
disp(%pi*a(1),"an(a0)")
disp("an(a1-->a5)")
for n=1:5
disp(2*a(n+1)*%pi)
end
disp("bn(b1-->b5)")
for n=1:5
disp(2*%pi*b(n+1))
end
//CTFS coefficients of a periodic signal
//x(t) =t
t = 0:0.01:1;
xt =2*%pi*t;
//
for k =0:6
  C(k+1,:) = exp(-sqrt(-1)*2*%pi*t*k);
  c(k+1) = xt*C(k+1,:)'/length(t);
  if(abs(c(k+1))<=0.01)
    c(k+1)=0;
  end
end
c =c';
c_conj = real(c(:))-sqrt(-1)*imag(c(:));
ck = [c_conj($:-1:1)',c(2:$)];
k = 0:6;
```

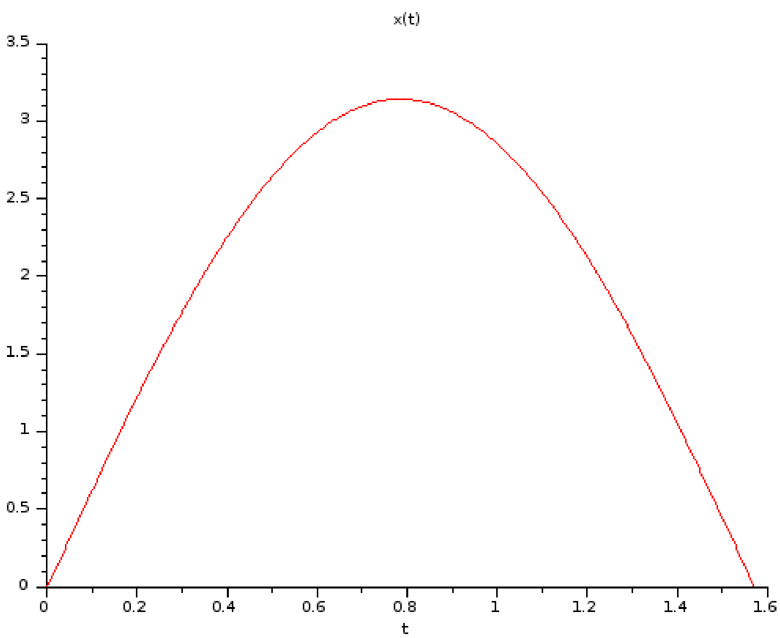

Figure 3.6: Complex Fourier Exponential Series

```
k = [-k($:-1:1),k(2:$)];
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot2d3('gnn',k,abs(ck))
poly1 = c.children(1).children(1);
poly1.thickness = 3;
title('|ck|')
xlabel('k')
```

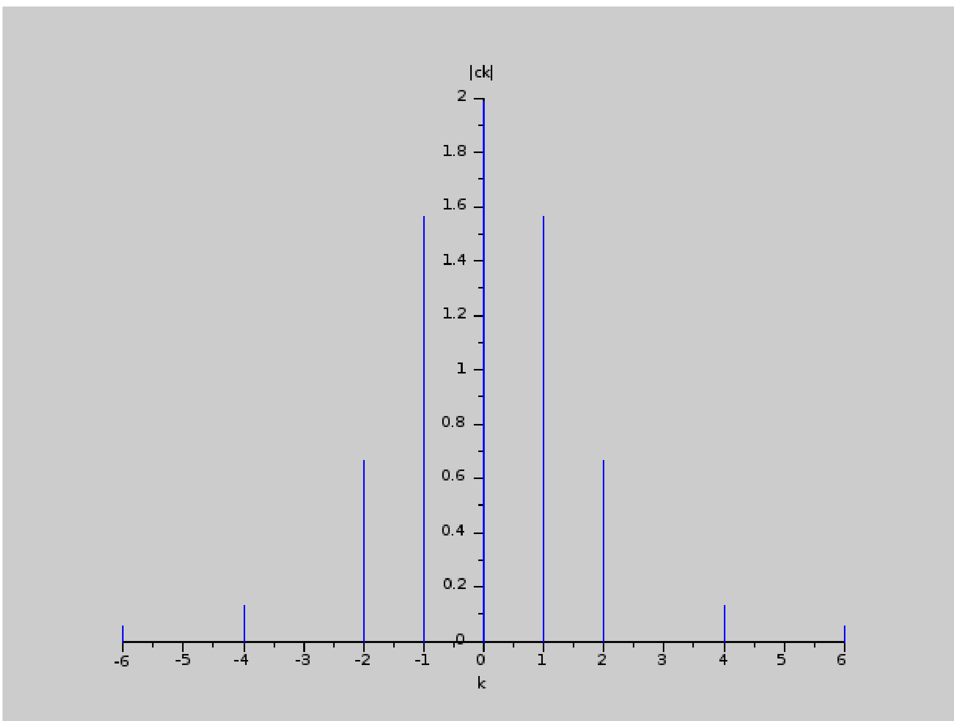

Figure 3.7: Complex Fourier Exponential Series

Scilab code Exa 3.17 Complex Fourier Exponential Series

```
//Scilab Code for Example 3.17 of Signals and systems by
//P.Ramakrishna Rao
//CTFS coefficients of a periodic signal
//x(wt) = Asin wt, 0<wt<pi
clear;
close;
clc;
A=3.14;
T =%pi;
T1 = T/2;
t = 0:0.01:T1;
Wo = 2*%pi/T;
xt =A*sin(Wo*t);
//
for k =0:6
  C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
  c(k+1) = xt*C(k+1,:)'/length(t);
  if(abs(c(k+1))<=0.01)
    c(k+1)=0;
  end
end
c =c';
c_conj = real(c(:))-sqrt(-1)*imag(c(:));
ck = [c_conj($:-1:1)',c(2:$)];
k = 0:6;
k = [-k($:-1:1),k(2:$)];
c = gca();
c.y_location = "origin";
c.x_location = "origin";
//c.data_bounds=[-2,0;2,2];
plot2d(t,xt,5)
poly1 = c.children(1).children(1);
poly1.thickness = 3;
title('x(t)')
xlabel('t')
```

```
figure(1);
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot2d3('gnn',k,abs(ck))
poly1 = c.children(1).children(1);
poly1.thickness = 3;
title('|ck|')
xlabel('k')
```

Scilab code Exa 3.18 Trignometric Series

```
//Scilab Code for Example 3.18 of Signals and systems by
//P.Ramakrishna Rao
//A=%pi or 3.14
clear;
clc;
//Trignometric Fourier Coefficients
a(1)=integrate('sin(w)','w',0,%pi);
for n=1:8
a(2*n+1)=integrate('sin(w+2*n*w)','w',0,%pi)+integrate('sin(w-2*w*n)','w',0,%pi);
end
for n=0:8
b(n+1)=0;
end
disp(abs(a(1)),"an(a0)");
disp("an(a1-->a8)");
n=1:8;
disp(2*a(n+1));
disp("bn(b1-->b8)");
n=1:8;
disp(b(n));
```

Chapter 4

The Continuous Time Fourier Transform

Scilab code Exa 4.1 Magnitude and Phase Spectra

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

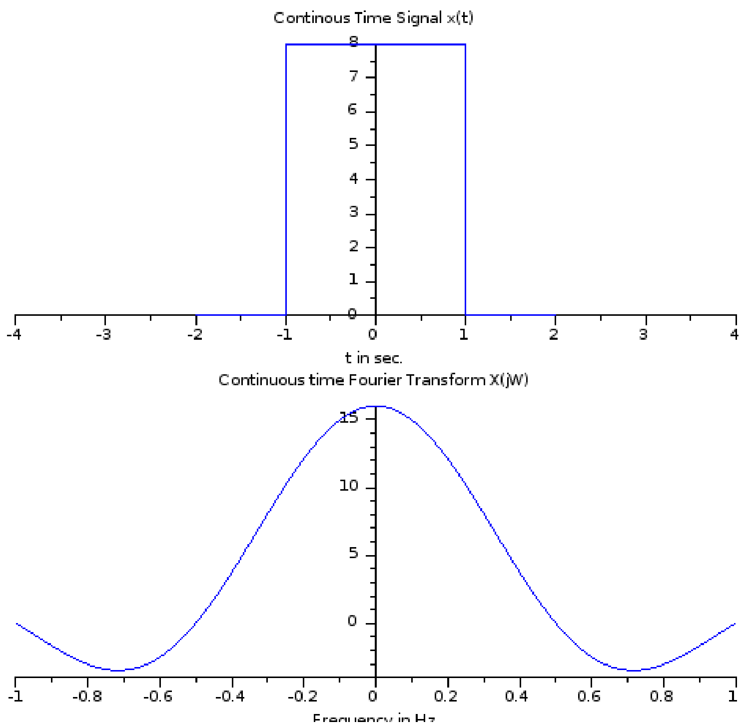

Figure 4.1: Magnitude and Phase Spectra

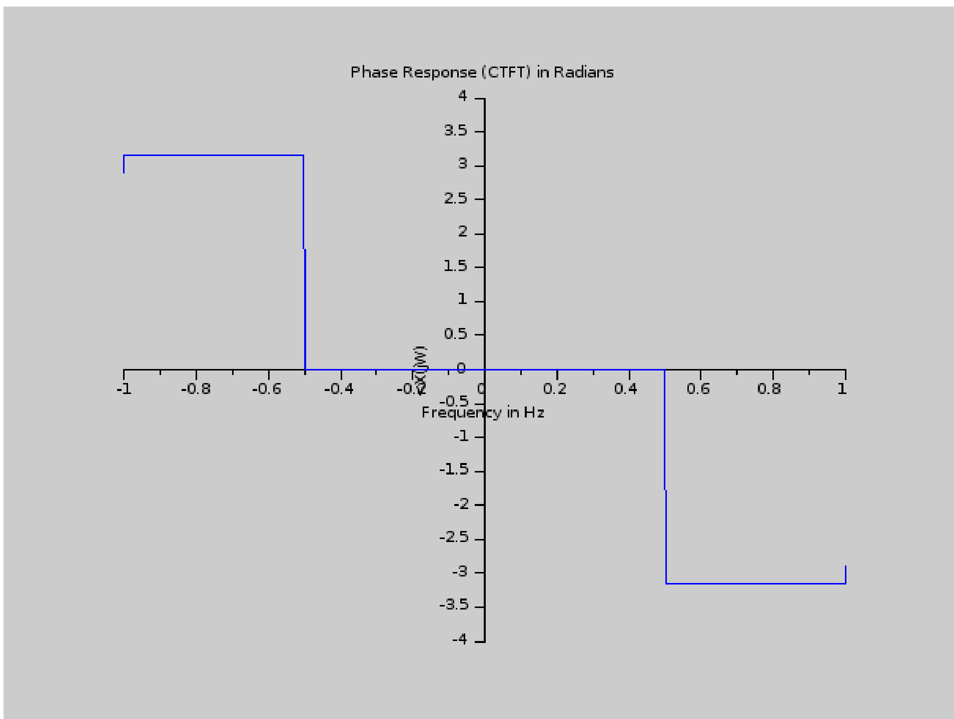

Figure 4.2: Magnitude and Phase Spectra

```
        end
end
Wmax = 2*%pi*1;        //Analog Frequency = 1Hz
K = 4;
k = 0:(K/1000):K;
W = k*Wmax/K;
xt = xt';
XW = xt* exp(-sqrt(-1)*t'*W) * Dt;
XW_Mag = real(XW);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -Wmax to Wmax
XW_Mag = [mtlb_fliplr(XW_Mag), XW_Mag(2:1001)];
[ XW_Phase ,db] = phasemag (XW);
XW_Phase =[-mtlb_fliplr(XW_Phase),XW_Phase(2:1001)];
// Plotting the Function
subplot(2,1,1);
a=gca();
a.data_bounds=[-4,0;4,2];
a.y_location="origin";
plot(t,xt);
xlabel('t in sec.');
title('Continous Time Signal x(t)');
// Plotting Magnitude Reponse of CTS
subplot(2,1,2);
a=gca();
a.y_location="origin";
plot(W/(2*%pi),XW_Mag);
xlabel('Frequency in Hz');
title('Continuous time Fourier Transform X(jW)');
// Plotting Phase Reponse of CTS
figure(1);
a = gca ();
a.y_location = "origin";
a.x_location = "origin";
plot (W/(2*%pi),-XW_Phase *%pi /180) ;
xlabel ( ' Frequency in Hz' );
ylabel ( '<X(jW) ' )
title ( ' Phase Response (CTFT) in Radians ' )
```

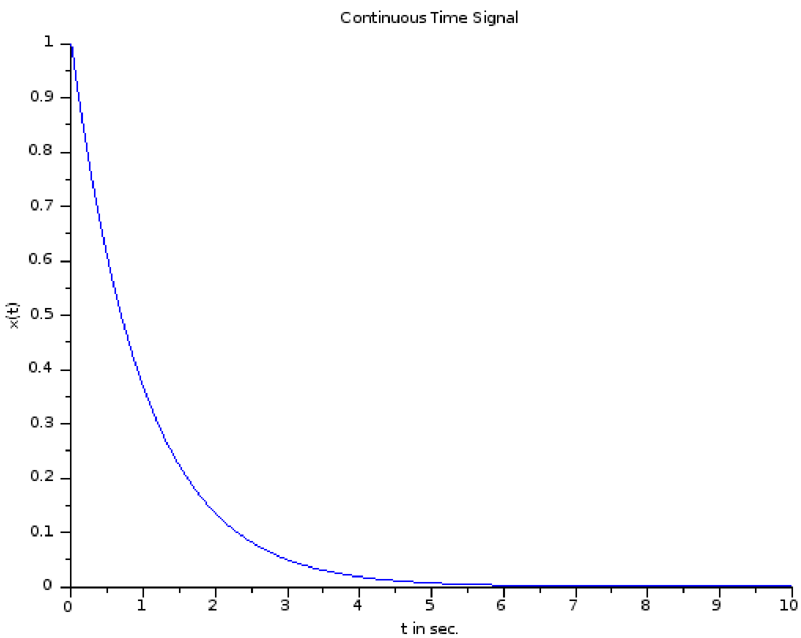

Figure 4.3: Magnitude and Phase Spectra

Scilab code Exa 4.2 Magnitude and Phase Spectra

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

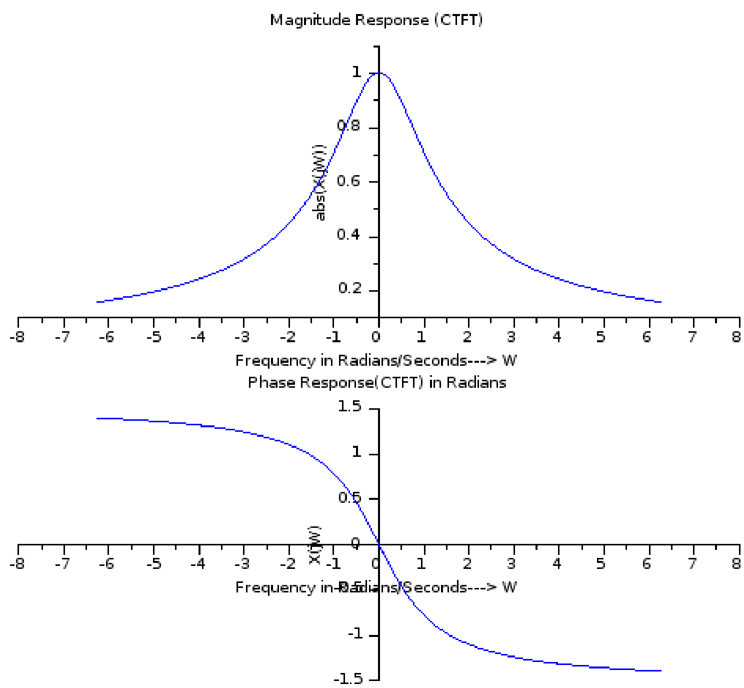

Figure 4.4: Magnitude and Phase Spectra

```
clc;
close;
// Analog Signal
A =1;     //Amplitude
T=1;
Dt = 0.005;
t = 0:Dt:10;
xt = A*exp(-t/T);
// Continuous-time Fourier Transform
Wmax = 2*%pi*1;         //Analog Frequency = 1Hz
K = 4;
k = 0:(K/1000):K;
W = k*Wmax/K;
XW = xt* exp(-sqrt(-1)*t'*W) * Dt;
XW_Mag = abs(XW);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
XW_Mag = [mtlb_fliplr(XW_Mag), XW_Mag(2:1001)];
[XW_Phase,db] = phasemag(XW);
XW_Phase = [-mtlb_fliplr(XW_Phase),XW_Phase(2:1001)
    ];
//Plotting Continuous Time Signal
a = gca();
a.y_location = "origin";
plot(t,xt);
xlabel('t in sec.');
ylabel('x(t)')
title('Continuous Time Signal')
figure
//Plotting Magnitude Response of CTS
subplot(2,1,1);
a = gca();
a.y_location = "origin";
plot(W,XW_Mag);
xlabel('Frequency in Radians/Seconds---> W');
ylabel('abs(X(jW))')
title('Magnitude Response (CTFT)')
//Plotting Phase Reponse of CTS
```

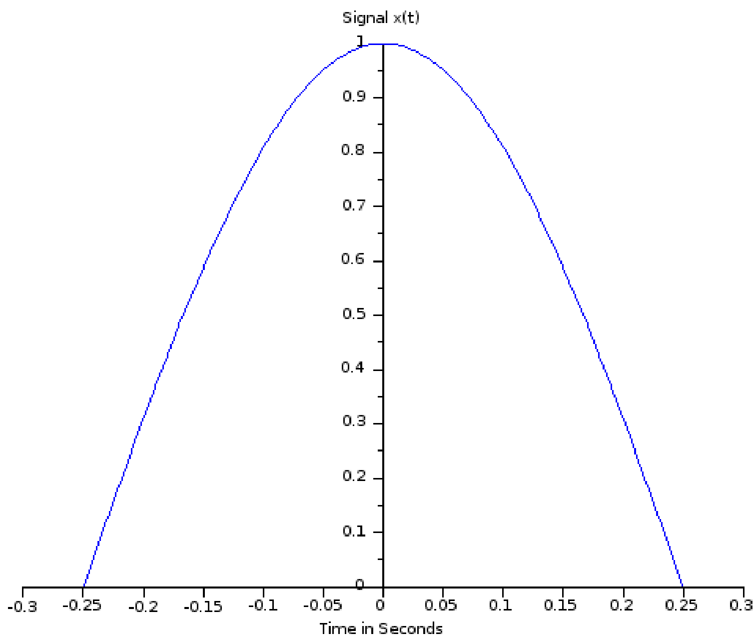

Figure 4.5: Magnitude and Phase Spectra

```
subplot(2,1,2);
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot(W,XW_Phase*%pi/180);
xlabel('                              Frequency in
    Radians/Seconds---> W');
ylabel('
                                                        <X
    (jW)')
title('Phase Response(CTFT) in Radians')
```

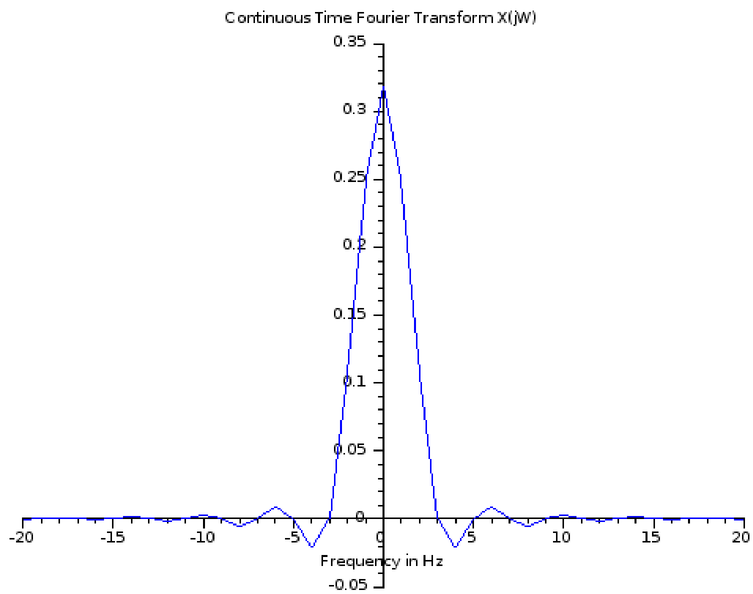

Figure 4.6: Magnitude and Phase Spectra

Scilab code Exa 4.3 Magnitude and Phase Spectra

```
//Scilab Code for Example 4.3 of Signals and systems
    by
//P.Ramakrishna Rao
//x(t)=A*cos w0*t*gate(t/T)
//T=1/2*fo
//fo=0.5 Hz
clear;
clc;
//Fourier Transform
A=1;
T=0.5;
fo=1/(2*T);
Wo=2*%pi*fo;
for f=-20:1:20;
X(f+21)=A*integrate('cos(Wo*t)*cos(2*%pi*f*t)','t'
    ,-0.25,0.25);
end
disp(X,'X(0)-->X(20)');
t=-0.25:0.01:0.25;
q=cos(Wo*t);
a = gca ();
a.y_location ="origin";
a.x_location ="origin";
plot(t,q);
xlabel ( 'Time in Seconds' );
title ('Signal x(t)');
figure(1);
a = gca ();
a.y_location ="origin";
a.x_location ="origin";
f=-20:1:20;
plot (f, X);
```

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

Scilab code Exa 4.4 Energy at output

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

Scilab code Exa 4.5 Fourier Transform

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

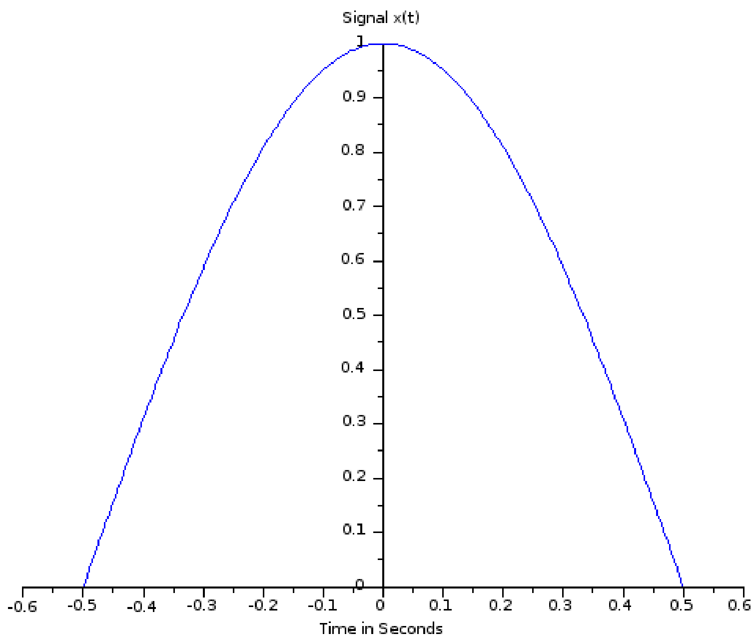

Figure 4.7: Fourier Transform

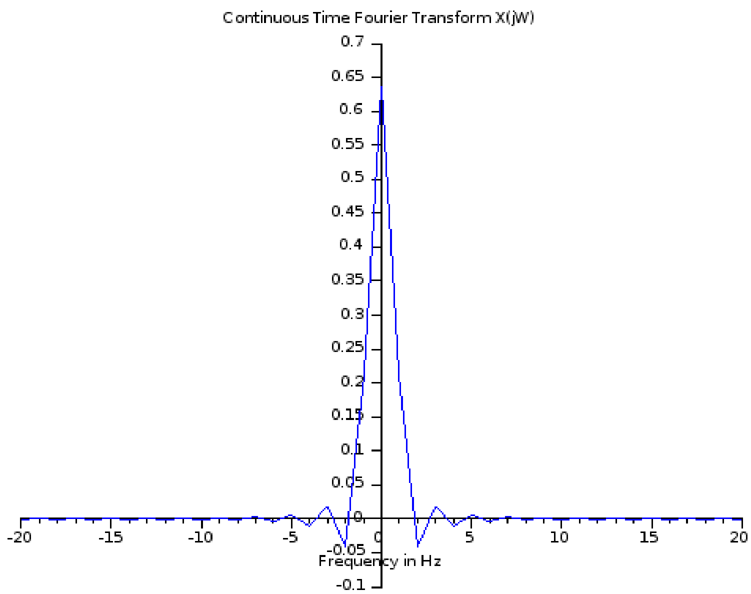

Figure 4.8: Fourier Transform

```
//x(t)=cos pi*t, |t|>0.5, zero otherwise
clear;
clc;
//Fourier Transform
for f=-20:1:20;
X(f+21)=integrate('cos(%pi*t)*cos(2*%pi*f*t)','t',-0.5,0.5);
end
disp(X,'X(0)-->X(20)');
t=-0.5:0.01:0.5;
q=cos(%pi*t);
a = gca ();
a.y_location ="origin";
a.x_location ="origin";
plot(t,q);
xlabel ( 'Time in Seconds' );
title ('Signal x(t)');
figure(1);
a = gca ();
a.y_location ="origin";
a.x_location ="origin";
f=-20:1:20;
plot (f, X);
xlabel ( 'Frequency in Hz ' );
title ('Continuous Time Fourier Transform X(jW)');
```

Scilab code Exa 4.6 Convolution

```
//Scilab Code for Example 4.6 of Signals and systems by
//P.Ramakrishna Rao
clear;
clc;
```

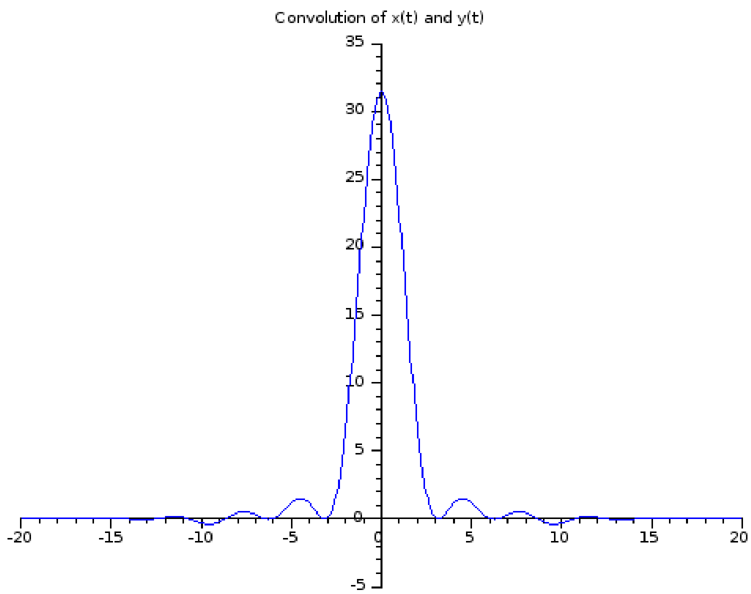

Figure 4.9: Convolution

```
close;
disp("X(w)=1/(a+jw)^2=1/(a+jw)*1/(a+jw) ");
disp("exp(-a*t)*u(t) <--> 1/(a+jw) ");
disp("therefore x(t)=convolution(2*sinc(2*t),(sinc t)^2");
a=2;
n=1;
for t=-10:0.1:10
y1(1,n)=(sinc(t))^2;
y2(1,n)=2*sinc(2*t);
n=n+1;
z=conv(y1,y2);
end
t=-20:0.1:20;
a=gca();
a.y_location ="origin";
a.x_location ="origin";
plot(t,z);
title('Convolution of x(t) and y(t)')
```

Scilab code Exa 4.7 Energy in Signal

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

Scilab code Exa 4.11 Fourier Transform

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

Scilab code Exa 4.14 Fourier Transform

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

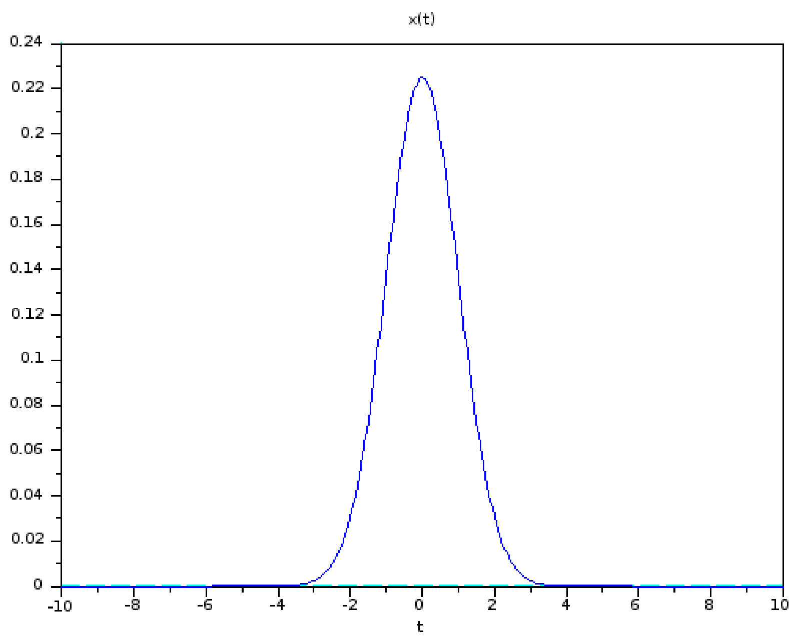

Figure 4.10: Fourier Transform

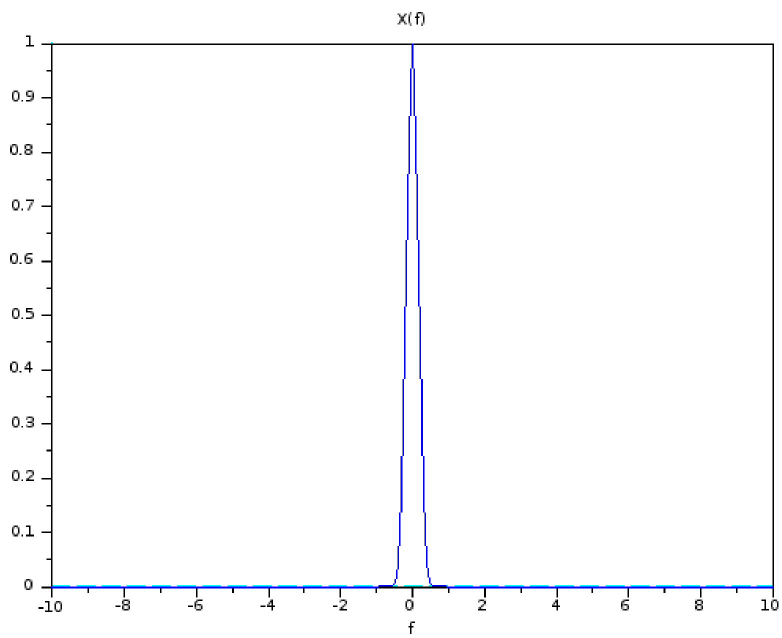

Figure 4.11: Fourier Transform

```
disp("Guassian pulse signal x(t)=(1/sqrt(2)*%pi)*exp(-a*t^2)");
disp("X(w)=integral(exp(-a*t^2)*exp(-%i*w*t)) w.r.t dt");
disp("d(X(w))/dw=-%i*w/(2*a)*integral(exp(-a*t^2)*exp(-%i*w*t))");
disp("d(X(w))/dw=-w*X(w)/2a");
disp("solving this we get X(w)=A*exp(-w^2/4a)")
disp("A=sqrt(%pi/a)");
d=gca()
plot(t,x);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('x(t)','t')
A=1;
f=t;
Xf=A*exp(-2*%pi^2*f^2);
figure(1);
d=gca()
plot(f,Xf);
poly1=d.children.children;
poly1.thickness=3;
poly1.foreground=2;
xtitle('X(f)','f')
```

Scilab code Exa 4.17 Fourier Transform

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

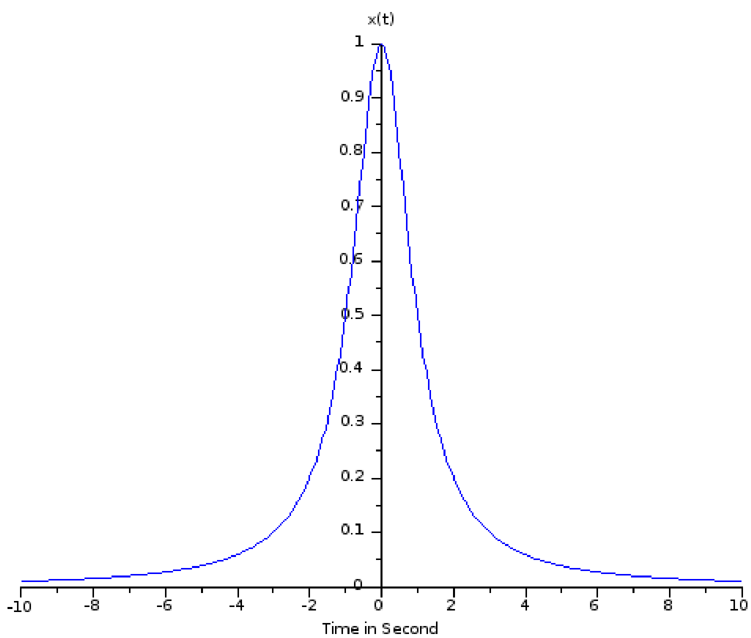

Figure 4.12: Fourier Transform

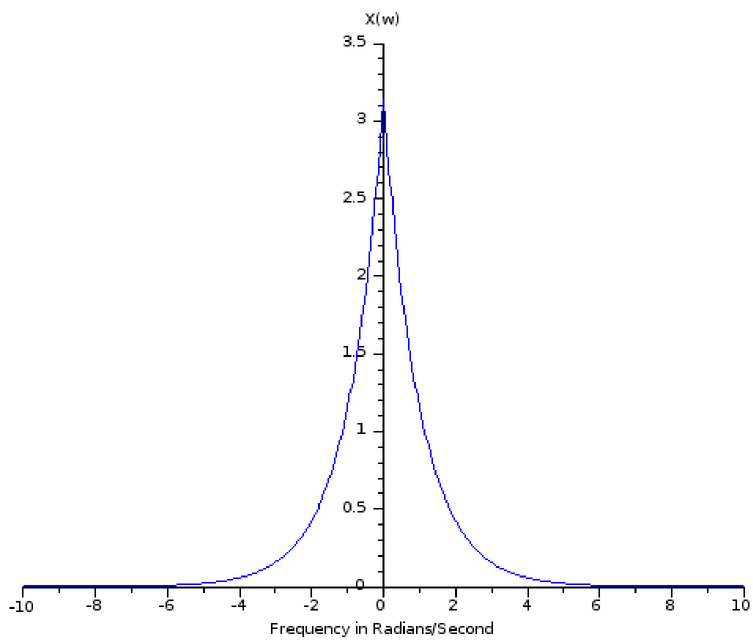

Figure 4.13: Fourier Transform

```
clear;
clc;
//Fourier Transform of x(t)
n=1;
for t=-10:0.1:10;
    x(1,n)=1/(1+t^2);
    n=n+1;
end
a = gca ();
a.y_location ="origin";
a.x_location ="origin";
t=-10:0.1:10;
plot(t,x);
xlabel ( 'Time in Second' );
title ('x(t)');
disp("By Duality Theorem Fourier Transform of x(t)
   is:");
disp("%pi*exp(-|w|)");
n=1;
for w=-10:0.1:10;
    X(1,n)=%pi*exp(-abs(w));
    n=n+1;
end
figure(1);
a = gca ();
a.y_location ="origin";
a.x_location ="origin";
w=-10:0.1:10;
plot(w,X);
xlabel ( 'Frequency in Radians/Second' );
title ('X(w)');
```

Chapter 5

Fourier Representation of Discrete Time Signals

Scilab code Exa 5.1 DTFT

```
//Scilab Code for Example 5.1 of Signals and systems
    by
//P.Ramakrishna Rao
//Discrete Time Fourier Transform of
//x[n]= (a^abs(n))  0<a<1
clear;
clc;
close;
// DTS Signal
a = 0.5;      //0<a<1
max_limit = 10;
n = -max_limit+1:max_limit-1;
x = a^abs(n);
// Discrete-time Fourier Transform
Wmax = 2*%pi;
K = 4;
k = 0:(K/1000):K;
W = k*Wmax/K;
XW = x* exp(-sqrt(-1)*n'*W);
```

```
XW_Mag = real(XW);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -
   Wmax to Wmax
XW_Mag = [mtlb_fliplr(XW_Mag), XW_Mag(2:1001)];
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',n,x);
xtitle('Discrete Time Sequence x[n] for a>0')
figure(1);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW_Mag);
title('Discrete Time Fourier Transform X(exp(jW))')
```

Scilab code Exa 5.2 DTFT and Spectra

```
//Scilab Code for Example 5.2 of Signals and systems
   by
//P.Ramakrishna Rao
//Discrete Time Fourier Transform of
//x[n]= 1 , 0=<n<=3
clear;
clc;
close;
// DTS Signal
N1 = 3;
n = 0:N1;
x = ones(1,length(n));
// Discrete-time Fourier Transform
```

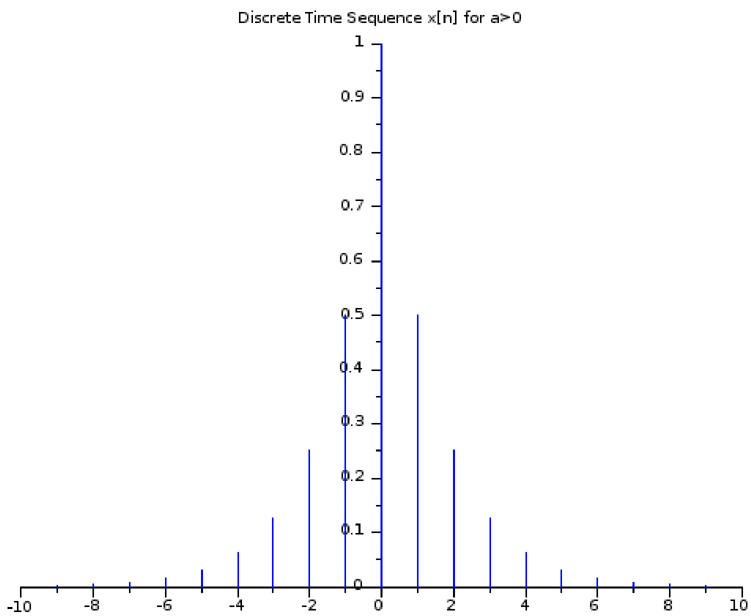

Figure 5.1: DTFT

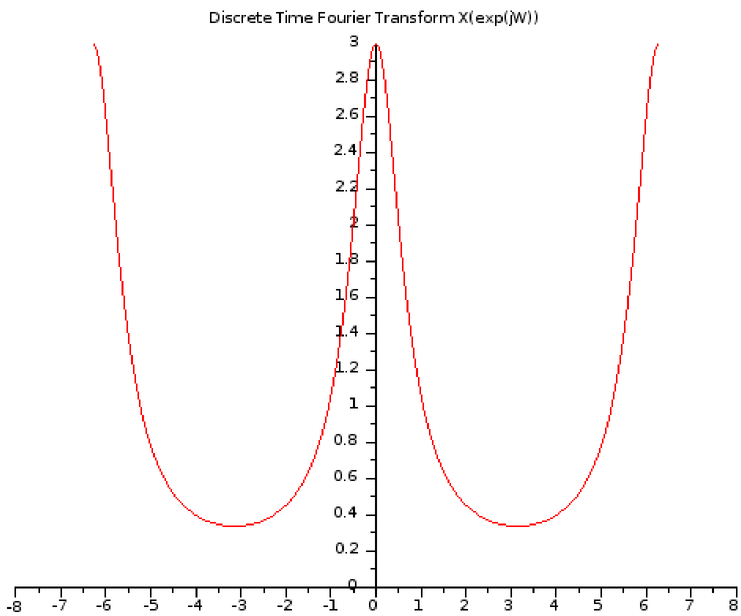

Figure 5.2: DTFT

```
Wmax = 2*%pi;
K = 4;
k = 0:(K/1000):K;
W = k*Wmax/K;
XW = x* exp(-sqrt(-1)*n'*W);
XW_Mag = real(XW);
[XW_Phase,db] = phasemag(XW);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -Wmax to Wmax
XW_Mag = [mtlb_fliplr(XW_Mag), XW_Mag(2:1001)];
XW_Phase = [-mtlb_fliplr(XW_Phase),XW_Phase(2:1001)];
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',n,x);
xtitle('Discrete Time Sequence x[n]')
figure(1);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,abs(XW_Mag));
title('Discrete Time Fourier Transform X(exp(jW))')
figure(2);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW_Phase);
title('Phase Response <(X(jW))')
```

Scilab code Exa 5.3 Inverse Fourier

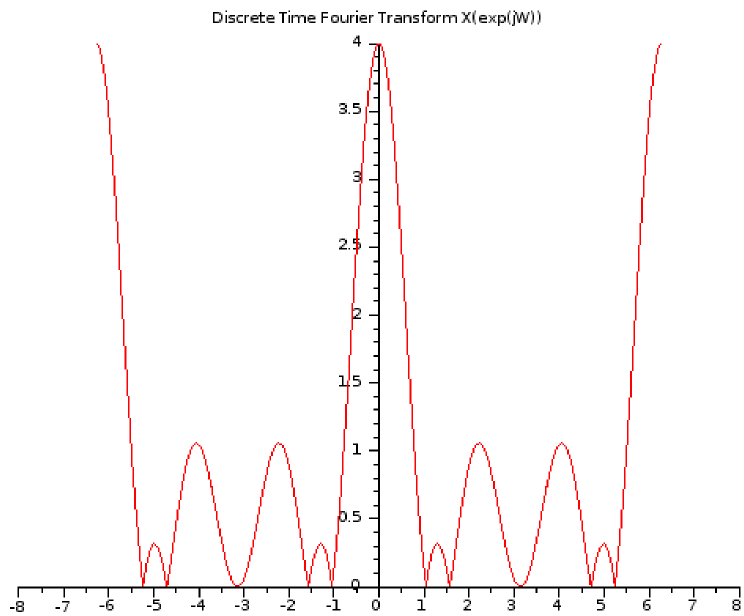

Figure 5.3: DTFT and Spectra

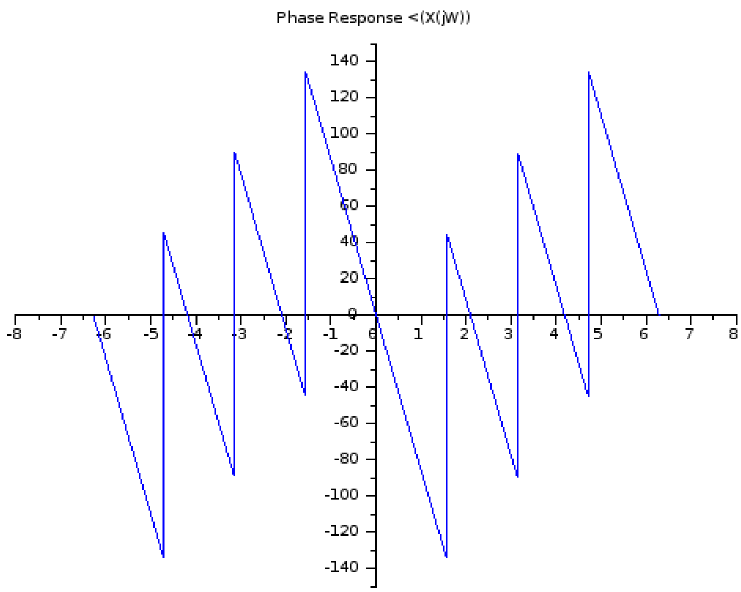

Figure 5.4: DTFT and Spectra

```
//Scilab Code for Example 5.3 of Signals and systems by
//P.Ramakrishna Rao
clear;
clc;
//Inverse Fourier Transform
w=-20:20;
X=4*(cos(w))^2;
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot(w,X);
title("X(e^jw)");
xlabel("Frequency in Radians/sec");
figure(1);
n=-50:50;
x=2*sinc(n)+sinc(n+2)+sinc(n-2);
a = gca ();
a.y_location ="origin";
a.x_location ="origin";
plot2d3(n,x);
title("x(n)");
xlabel("Time in sec");
```

Scilab code Exa 5.4 Inverse Fourier

```
//Scilab Code for Example 5.4 of Signals and systems by
//P.Ramakrishna Rao
//Discrete Time Fourier Transform of
// X(e^j*w)=2*%pi*delta(w)
```

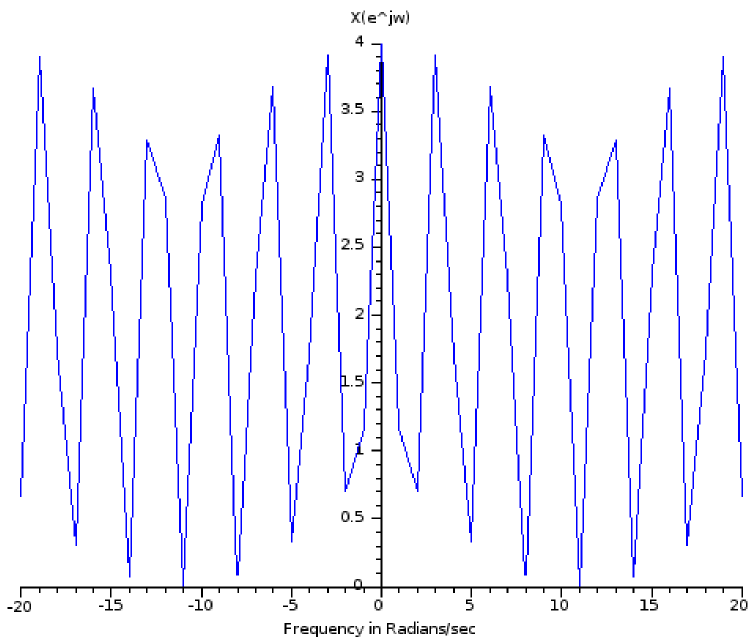

Figure 5.5: Inverse Fourier

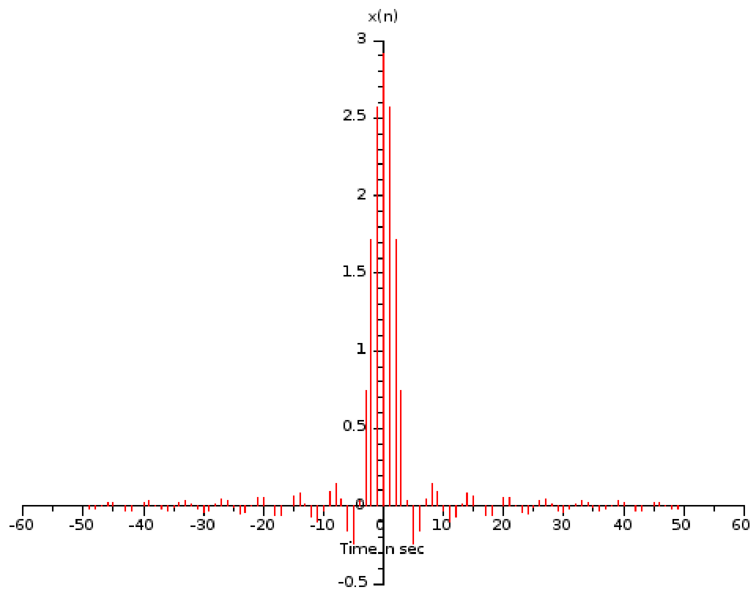

Figure 5.6: Inverse Fourier

```
clear;
clc;
close;
N = 1;
N1 = -3*N:3*N;
xn = [zeros(1,N-1),1];
x = [1 xn xn xn xn xn xn];
ak = 1/N;
XW = 2*%pi*ak*ones(1,2*N);
Wo = 2*%pi/N;
n  = -N:N-1;
W = Wo*n;
a = gca();
a.y_location ="middle";
a.x_location ="origin";
plot2d3('gnn',W,XW,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('W');
title('DTFT of Periodic Impulse Train')
figure(1);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',N1,x,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('n');
title('Periodic Impulse Train x(n)')
```

Scilab code Exa 5.5 Fourier series

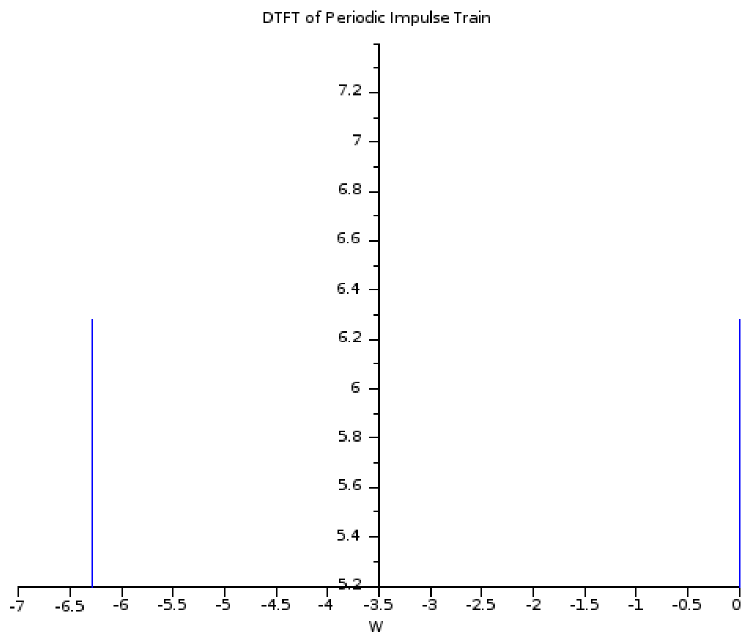

Figure 5.7: Inverse Fourier

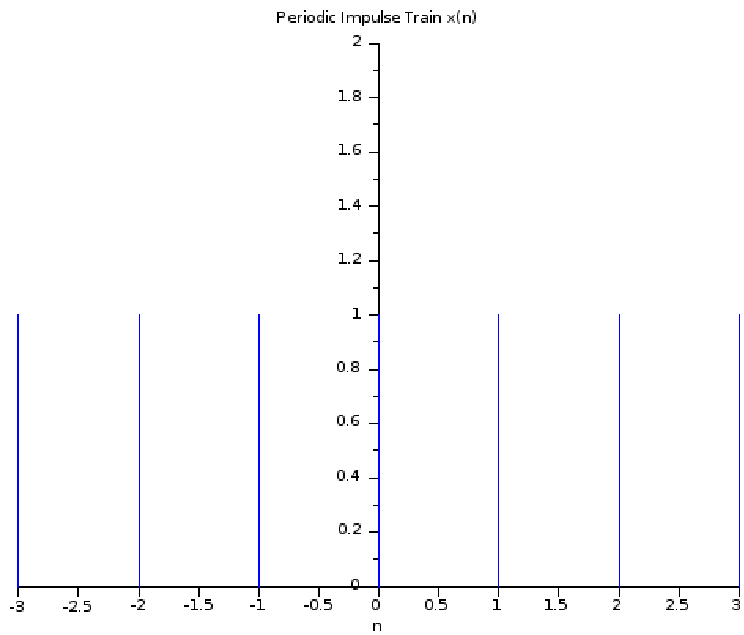

Figure 5.8: Inverse Fourier

```
//Scilab Code for Example 5.5 of Signals and systems by
//P.Ramakrishna Rao
clear;
clc;
x=[1,2,3,2,1];
q=0;
for n=1:5
    q=x(n)+q;
end
disp(q,'(a) X(e^j*0)');
q=0;
for n=-2:2
    q=((-1)^n)*x(n+3)+q;
end
disp(q,'(c) X(e^j*pi)');
disp('(d) X(e^j*pi)=2*pi*x(0)');
disp(2*%pi*x(3));
q=0;
for n=-2:2
    q=(x(n+3))^2+q;
end
disp(q*2*%pi,'(e) |X(e^j*w)|^2');
```

Scilab code Exa 5.9.i DTFT

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

```
x1=a1^n;
x2=a2^n;
w=2;
n=0:100;
z=(exp(-%i*w*n));
for n=0:33;
    X(n+1)=z(n+1)*x1(3*n+1);
    q=X(n+1)+q;
end
disp(q,'Y1(e^2j) at a=0.5');
for n=0:33;
    X(n+1)=z(n+1)*x2(3*n+1);
    q=X(n+1)+q;
end
disp(q,'Y1(e^2j) at a=-0.5');
```

Scilab code Exa 5.9.ii DTFT

```
//Scilab Code for Example 5.9(ii) of Signals and
   systems by
//P.Ramakrishna Rao
clc;
clear;
q=0;
a1=0.5
a2=-0.5;
n=1:101;
x1=a1^n;
x2=a2^n;
w=2;
n=0:100;
z=(exp(-%i*w*n));
for n=0:33;
    X(n+1)=z(n+1)*x1(n+1);
    q=X(n+1)+q;
```

```
end
disp(q,'Y2(e^2j) at a=0.5');
for n=0:33;
    X(n+1)=z(n+1)*x2(n+1);
    q=X(n+1)+q;
end
disp(q,'Y2(e^2j) at a=-0.5');
```

Scilab code Exa 5.9.iii DTFT

```
//Scilab Code for Example 5.9(iii) of Signals and
    systems by
//P.Ramakrishna Rao
clc;
clear;
q=0;
a1=0.5
a2=-0.5;
n=1:101;
x1=a1^n;
x2=a2^n;
w=2;
n=0:100;
z=(exp(-%i*w*n));
for n=0:33;
    X(n+1)=z(n+1)*x1(n+1)*cos(0.4*%pi*n);
    q=X(n+1)+q;
end
disp(q,'Y3(e^2j) at a=0.5');
for n=0:33;
    X(n+1)=z(n+1)*x2(n+1)*cos(0.4*%pi*n);
    q=X(n+1)+q;
end
disp(q,'Y3(e^2j) at a=-0.5');
```

Scilab code Exa 5.11 Inverse Fourier

```
//Scilab Code for Example 5.11 of Signals and
   systems by
//P.Ramakrishna Rao
clear;
clc;
wc=1;
y=1;
for n=-%pi:%pi/80:%pi
    if n<-wc | n>wc then
        X(1,y)=1;
        y=y+1;
    else X(1,y)=0;
        y=y+1;
    end
end
n=-%pi:%pi/80:%pi;
a = gca ();
a.y_location ="origin";
a.x_location ="origin";
plot(n,X);
xlabel ( 'Frequency in Radians/Seconds' );
title ('X(e^jw))    at    Wc=1');
A=1/%pi;
for k=-10:10
    x(k+11)=A*integrate('cos(w*k)','w',wc,%pi);
end
figure(1);
k=-10:10;
a = gca ();
a.y_location ="origin";
a.x_location ="origin";
plot2d3(k,x);
```

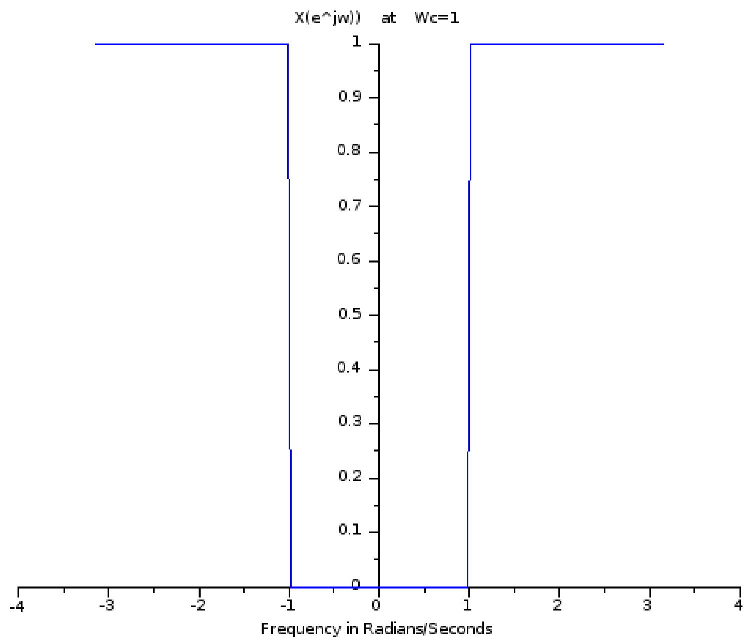

Figure 5.9: Inverse Fourier

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

Scilab code Exa 5.13 Circular Convolution

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

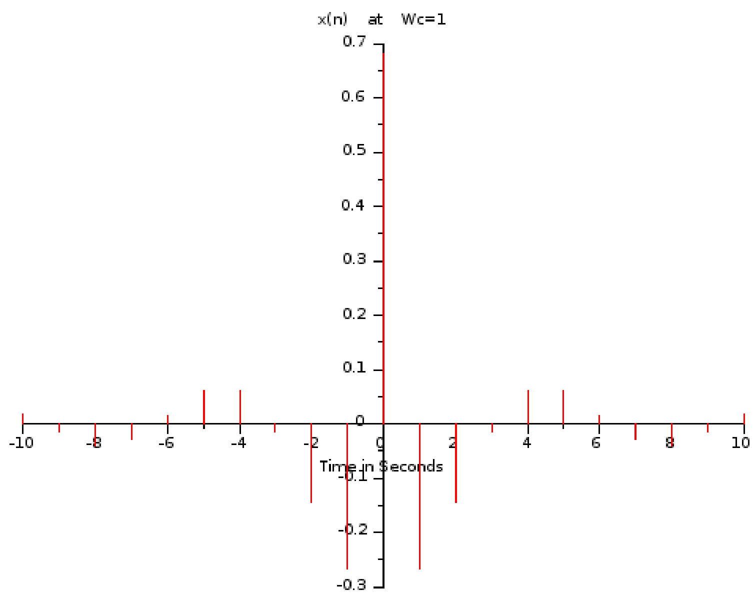

Figure 5.10: Inverse Fourier

```
clear;
//First Causal sequence
x=[1,4,3,2;2,1,4,3;3,2,1,4;4,3,2,1]
//Second Sequence
y=[4;3;2;1]
//Conolution
z=x*y;
disp(z,'Convolution Of x & y is:')
```

Scilab code Exa 5.14 Circular Convolution

```
//Scilab Code for Example 5.14 of Signals and
   systems by
//P.Ramakrishna Rao
//Circular Convolution
clear;
clc;
close;
//First Causal sequence
x=[1,1,1,1;1,-%i,-1,%i;1,-1,1,-1;1,%i,-1,-%i]
y1=[1;2;3;4];
y2=[4;3;2;1];
X=x*y1;
disp(X,'Vector X(0)-->X(3)')
Y=x*y2;
disp(Y,'Vector Y(0)-->Y(3)')
for n=1:4;
    Z(n,1)=X(n,1)*Y(n,1);
end
q=4*x^-1;
disp(q,'IDFT matrix ');
z=0.25*q*Z;
disp(z,'IDFT of Vector Z(0)-->Z(3)')
```

Scilab code Exa 5.16 DTFT

```
//Scilab Code for Example 5.16 of Signals and
    systems by
//P.Ramakrishna Rao
clc;
clear;
q=0;
x=[1,2,3,4];
w=0;
n=0:3;
z=(exp(-%i*w*n));
for n=0:3
    X(n+1)=z(n+1)*x(n+1);
    q=X(n+1)+q;
end
disp(q,'X(0)->zeroth DFT coefficient');
q=0;
w=%pi/2;
n=0:3;
z=(exp(-%i*w*n));
for n=0:3
    X(n+1)=z(n+1)*x(n+1);
    q=X(n+1)+q;
end
disp(q,'X(1)->first DFT coefficient');
q=0;
w=%pi;
n=0:3;
z=(exp(-%i*w*n));
for n=0:3
    X(n+1)=z(n+1)*x(n+1);
    q=X(n+1)+q;
end
```

```
disp(ceil(q),'X(2)->second DFT coefficient');
q=0;
w=3*%pi/2;
n=0:3;
z=(exp(-%i*w*n));
for n=0:3
    X(n+1)=z(n+1)*x(n+1);
    q=X(n+1)+q;
end
disp(q,'X(3)->third DFT coefficient');
```

Scilab code Exa 5.18 DFT coefficients

```
//Scilab Code for Example 5.18 of Signals and
   systems by
//P.Ramakrishna Rao
//Given signal x(n)
clear;
clc;
x=[1,2,3,4];
X=fft(x);
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

```
//Scilab Code for Example 6.1 of Signals and systems by
//P.Ramakrishna Rao
//Sampling of signal
clc;
clear f n X X_delta w;
fs=200;
for f=-200:200
    X(f+201)=5*[delta(f-75)+delta(f+75)];
end
figure(1);
f=-200:200;
plot2d3(f,X,-2);
title('X(f)');
xlabel('---->  f');
w=1;
n=-1;
    for f=-275:275
    X_delta1(f+276)=fs*5*[delta(f-n*fs-75)+delta(f-n
```

```
        *fs+75)];
    end
n=n+1;
    for f=-275:275
    X_delta2(f+276)=fs*5*[delta(f-n*fs-75)+delta(f-n*fs+75)];
end
n=n+1;
    for f=-275:275
    X_delta3(f+276)=fs*5*[delta(f-n*fs-75)+delta(f-n*fs+75)];
end
X_delta=X_delta1+X_delta2+X_delta3;
figure(2);
f=-275:275;
plot2d3(f,X_delta,-2);
title('X_delta(f) at fs=200');
xlabel('---> f');
```

Scilab code Exa 6.2 Sampling and Reconstruction

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

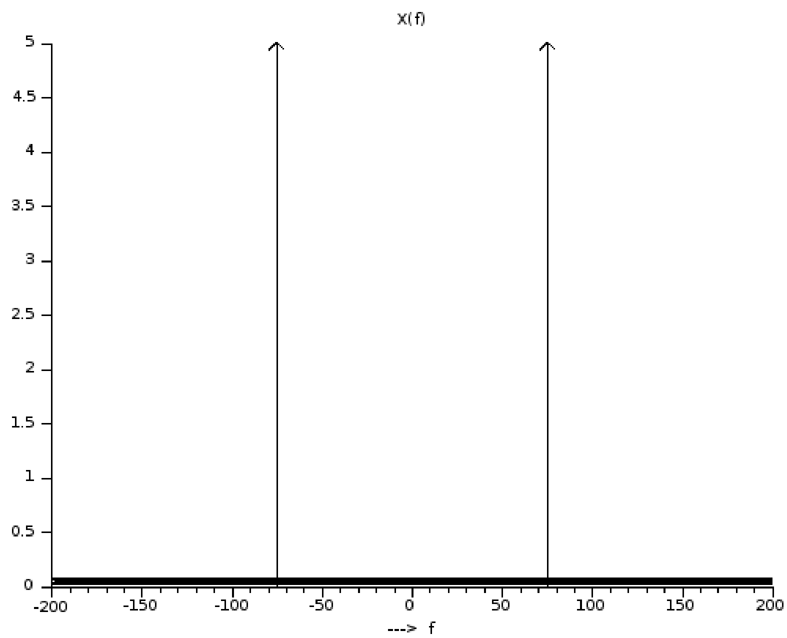

Figure 6.1: Sampling and Reconstruction

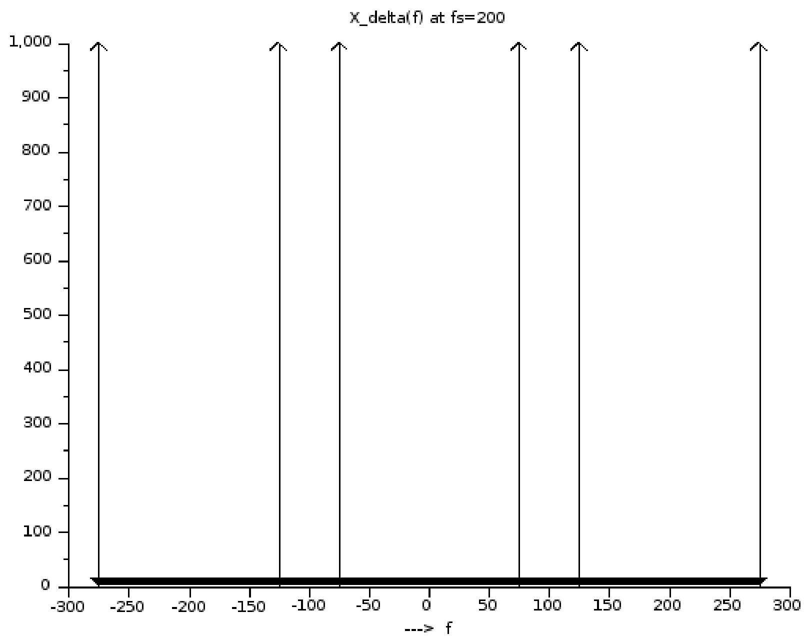

Figure 6.2: Sampling and Reconstruction

```
end
figure(1);
f=-200:200;
plot2d3(f,X,-2);
title('X(f)');
xlabel('--->  f');
w=1;
n=-2;
    for f=-175:175
    X_delta1(f+176)=fs*5*[delta(f-n*fs-75)+delta(f-n*fs+75)];
    end
n=n+1;
    for f=-175:175
    X_delta2(f+176)=fs*5*[delta(f-n*fs-75)+delta(f-n*fs+75)];
end
n=n+1;
    for f=-175:175
    X_delta3(f+176)=fs*5*[delta(f-n*fs-75)+delta(f-n*fs+75)];
end
n=n+1;
    for f=-175:175
    X_delta4(f+176)=fs*5*[delta(f-n*fs-75)+delta(f-n*fs+75)];
end
n=n+1;
    for f=-175:175
    X_delta5(f+176)=fs*5*[delta(f-n*fs-75)+delta(f-n*fs+75)];
end
X_delta=X_delta1+X_delta2+X_delta3+X_delta4+X_delta5;
figure(2);
f=-175:175;

plot2d3(f,X_delta,-2);
```

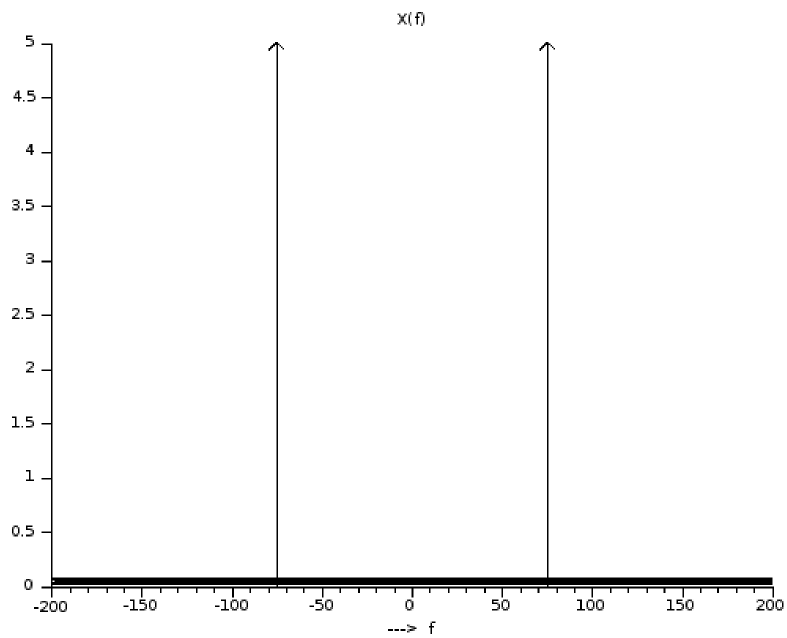

Figure 6.3: Sampling and Reconstruction

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

Scilab code Exa 6.3 Minimum Number of samples

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

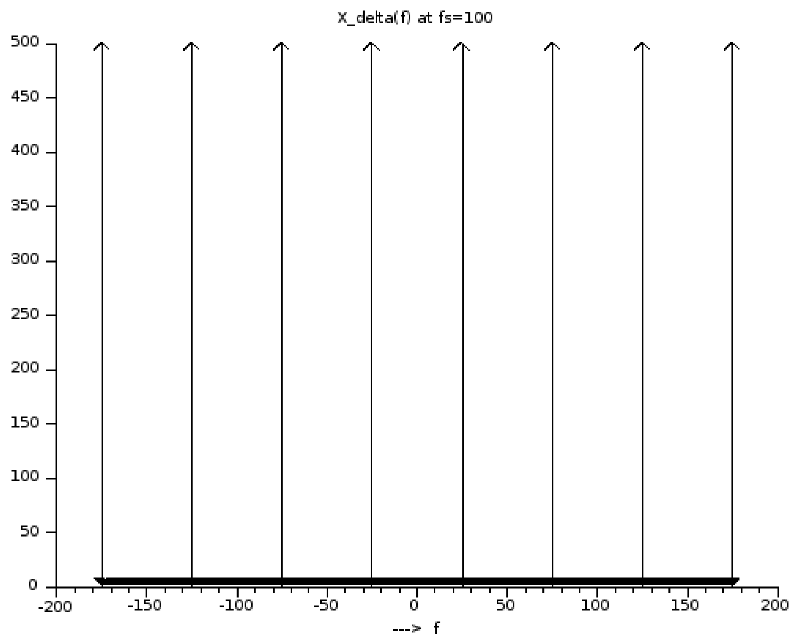

Figure 6.4: Sampling and Reconstruction

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

Scilab code Exa 6.4 Minimum Sampling Frequency

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

Scilab code Exa 6.5 LPF output

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

```
clear;

for f=-100:100
    X(f+101)=delta(f+100)+delta(f-100)+3*[delta(f+90)+delta(f-90)];
end
f=-100:100;
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d3(f,X,-2);
title('Spectrum of x(t)');
xlabel('--> f');
fs=150;
n=-1;f-n*fs
    for f=-275:275
    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)];
    end
n=n+1;
    for f=-275:275
    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)];
end
n=n+1;
    for f=-275:275
    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)];
end
X_delta=X_delta1+X_delta2+X_delta3;
figure(2);
f=-275:275;
plot2d3(f,X_delta,-2);
title('X_delta(f) at fs=150');
xlabel('---> f');
```

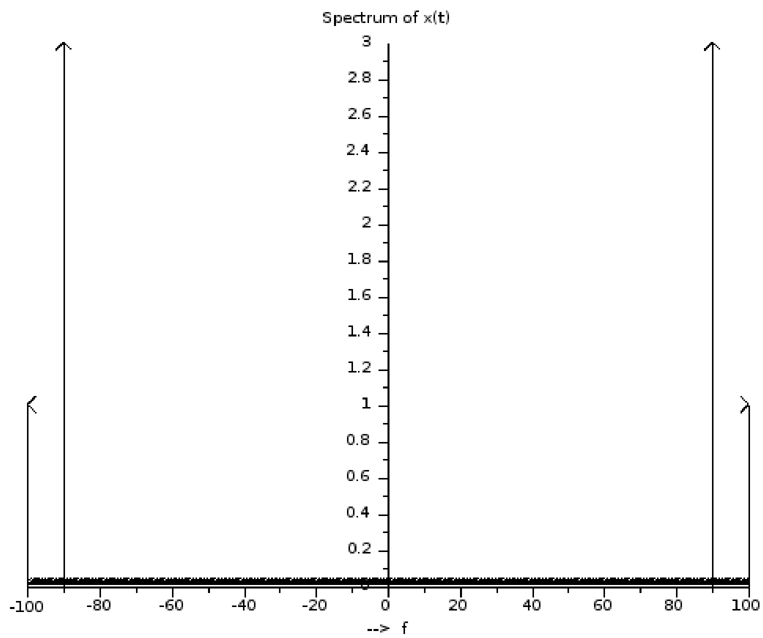

Figure 6.5: LPF output

Scilab code Exa 6.6 Sampling Frequency

```
//Scilab Code for Example 6.6 of Signals and systems
    by
//P.Ramakrishna Rao
//Sampling Frequency / Nyquist Rate
clc;
clear;
```

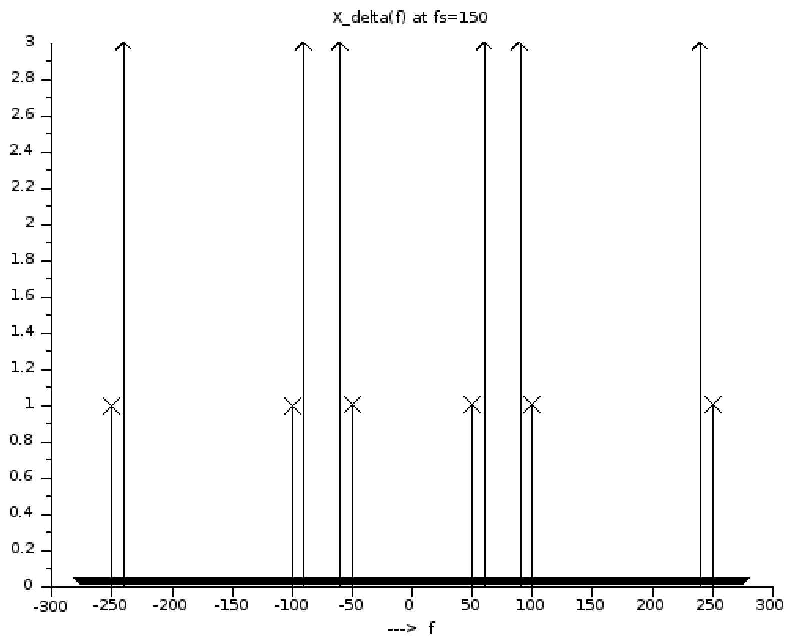

Figure 6.6: LPF output

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

Scilab code Exa 6.8 Minimum Sampling Frequency

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

Chapter 7

Systems

Scilab code Exa 7.1 Properties of System

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

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

Scilab code Exa 7.3 System Properties

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

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

Scilab code Exa 7.4 System Properties

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

```
disp('For a delay (T) of 2 seconds');
disp('At t=3 seconds:');
t=3;
a=x(1,2*t-2);
b=y(1,t-2);
c=x(1,2*t-4);
disp(a,'x(2t-T):');
disp(b,'is not equal to y(2t-T):');
disp(c,'while x(2t-2*T):');
disp('Hence the system is Time variant');
```

Scilab code Exa 7.5 System Properties

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

```
end
disp(q,'This input gives the output:');
y=b1*y1+b2*y2;
disp(y,'For the system to be linear the output
    should be:');
disp('Hence the system is linear');
disp('For a delay (n0) of 2 seconds');
disp('At n=3 seconds:');
t=3;
a=x(1,2*n-2);
b=y(1,n-2);
c=x(1,2*n-4);
disp(a,'x(2n-n0):');
disp(b,'is not equal to y(2n-n0):');
disp(c,'while x(2n-2*n0):');
disp('Hence the system is Time variant');
```

Scilab code Exa 7.6 System Properties

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

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

Scilab code Exa 7.7 Properties of System

```
//Scilab Code for Example 7.7 of Signals and systems
    by
//P.Ramakrishna Rao
clc;
clear x y1 y y2 q t n;
clear;
//y(t)=n*x(n)
disp('y(n) depends only upon present inputs');
disp('(i)    Hence the system is Dynamic');
x1=[1,3,5,3,2,5,3,9];//random variable
x2=[2,4,6,4,2,4,2,1];
for n=1:4
    y1(1,n)=n*x1(n);
    y2(1,n)=n*x2(n);
end
b1=2;
b2=3;
x=b1*x1+b2*x2;
disp(x,'The input to the system is:');
for n=1:4
    q(1,n)=n*x(n);
end
disp(q,'This input gives the output:');
y=b1*y1+b2*y2;
disp(y,'For the system to be linear the output
   should be:');
disp('(ii)    Hence the system is linear');
disp('For a delay (n0) of 2 seconds');
```

```
disp('At n=3 seconds:');
t=3;
a=x(1,n-2);
b=y(1,n-2);
c=2*x(1,n-2);
disp(a,'x(n-n0):');
disp(b,'is not equal to y(n-n0):');
disp(c,'while (n-n0)*x(n-n0):');
disp('(iii)    Hence the system is Time variant');
```

Scilab code Exa 7.8 Properties of System

```
//Scilab Code for Example 7.8 of Signals and systems
    by
//P.Ramakrishna Rao
clc;
clear x y1 y y2 q t n;
clear;
//y(t)=sum(x(n)),n-2<=n=<n+2
x1=[1,3,5,3,0,0,0,0,0,0];//random variable
x2=[2,4,6,4,0,0,0,0,0,0];
for n=1:4
    y1(1,n)=x1(n)+x1(n+1)+x1(n+2)+x1(n+3)+x1(n+4);
    y2(1,n)=x2(n)+x2(n+1)+x2(n+2)+x2(n+3)+x2(n+4);
end
b1=2;
b2=3;
x=b1*x1+b2*x2;
disp(x,'The input to the system is:');
for n=1:4
    q(1,n)=x(n)+x(n+1)+x(n+2)+x(n+3)+x(n+4);
end
disp(q,'This input gives the output:');
y=b1*y1+b2*y2;
disp(y,'For the system to be linear the output
```

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

Scilab code Exa 7.9 Properties of System

```
//Scilab Code for Example 7.9 of Signals and systems
    by
//P.Ramakrishna Rao
clc;
clear x y1 y y2 q t n;
clear;
//y(t)=exp|x(n)|
x1=[1,3,5,3];//random variable
x2=[2,4,6,4];
for n=1:4
    y1(1,n)=exp(abs(x1(n)));
    y2(1,n)=exp(abs(x2(n)));
end
b1=2;
b2=3;
x=b1*x1+b2*x2;
disp(x,'The input to the system is:');
for n=1:4
    q(1,n)=exp(abs(b1*(x1(n))+b2*(x2(n))));
end
disp(q,'This input gives the output:');
y=b1*y1+b2*y2;
disp(y,'For the system to be linear the output
   should be:');
disp('(ii)    Hence the system is not linear');
disp('For a delay (n0) of 2 seconds');
disp('At n=3 seconds:');
n=4;
a=exp(abs(x1(n-2)));
b=y1(1,n-2);
```

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

Scilab code Exa 7.10 Output of an LTI

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

Scilab code Exa 7.11 Convolution

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

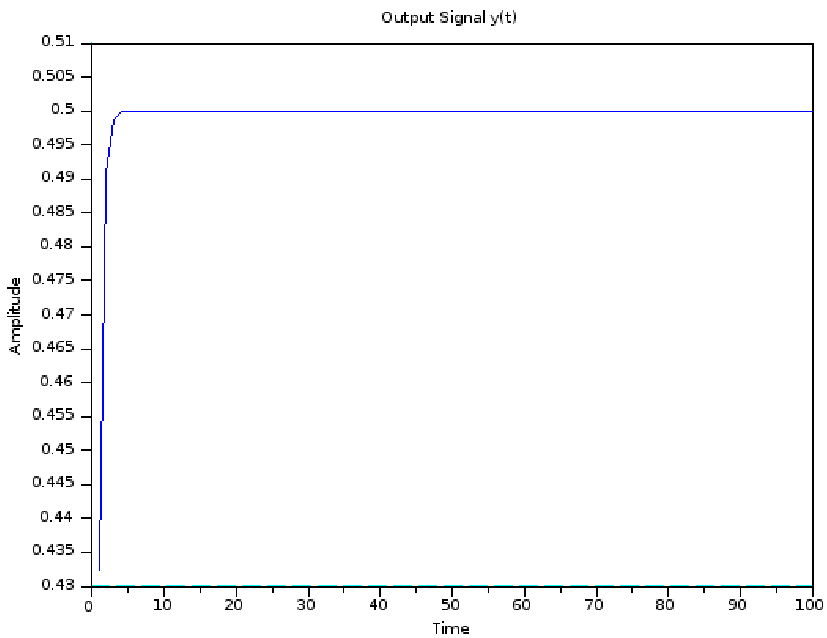

Figure 7.1: Output of an LTI

```
n=-2:2;
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot2d2(n,x,2);
title('x(t)')
xlabel('t')
figure(1);
n=-2:2;
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot2d2(n,y,5);
title('y(t)')
xlabel('t')
z=conv(x,y);
figure(2);
n=-3:5;
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot(n,z,2);
title('Convoluted signal      z(t)')
xlabel('t')
```

Scilab code Exa 7.14 Impulse and Step response

```
//Scilab Code for Example 7.14 of Signals and
   systems by
//P.Ramakrishna Rao
//Plotting the impulse and step responses
clc;
clear;
```

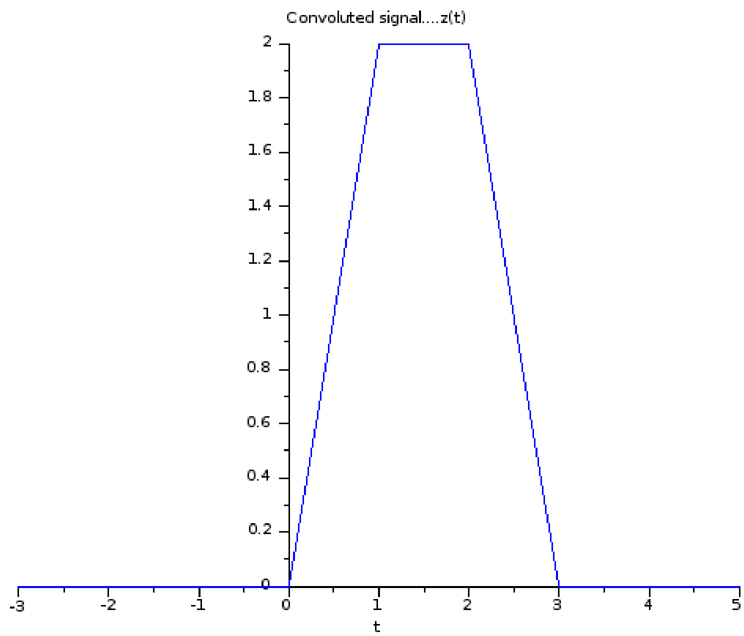

Figure 7.2: Convolution

```
syms s t R C;
Y1=(1/(R*C))/(s+1/(R*C));
disp(Y1,'Laplace Transform Of differential Equation
    is:')
y11=ilaplace(Y1,s,t);
disp(y11,'The Impulse Response of the System is:');
for k=0:10;
    y1(k+1)=exp(-k);
end
k=0:10;
plot(k,y1);
title('System Response to impulse input');
Y2=(1/(R*C))/(s^2+s/(R*C));
disp(Y2,'Laplace Transform Of differential Equation
    is:')
y22=ilaplace(Y2,s,t);
disp(y22,'The Step Response of the System is:');
for k=0:10;
    y2(k+1)=1-exp(-k);
end
figure(1);
k=0:10;
plot(k,y2);
title('System Response to Step input');
```

Scilab code Exa 7.15 Impulse and Step Response

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

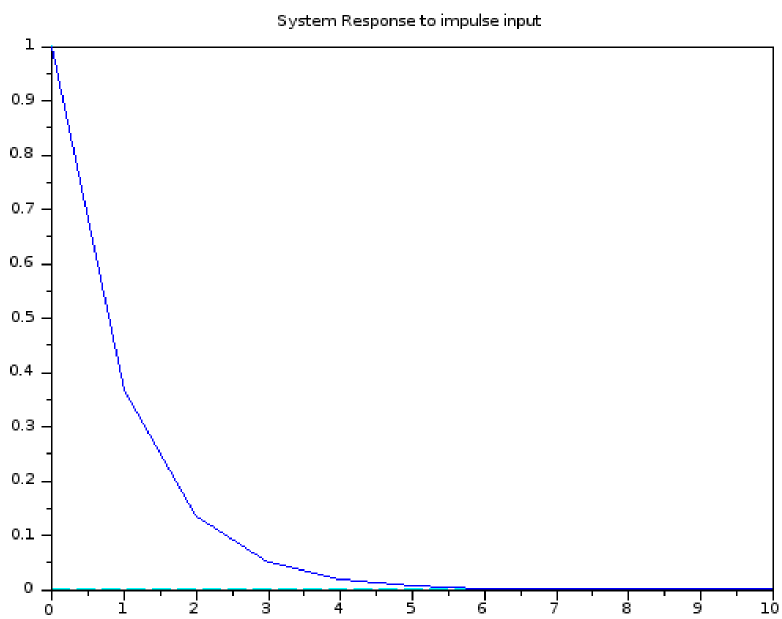

Figure 7.3: Impulse and Step response

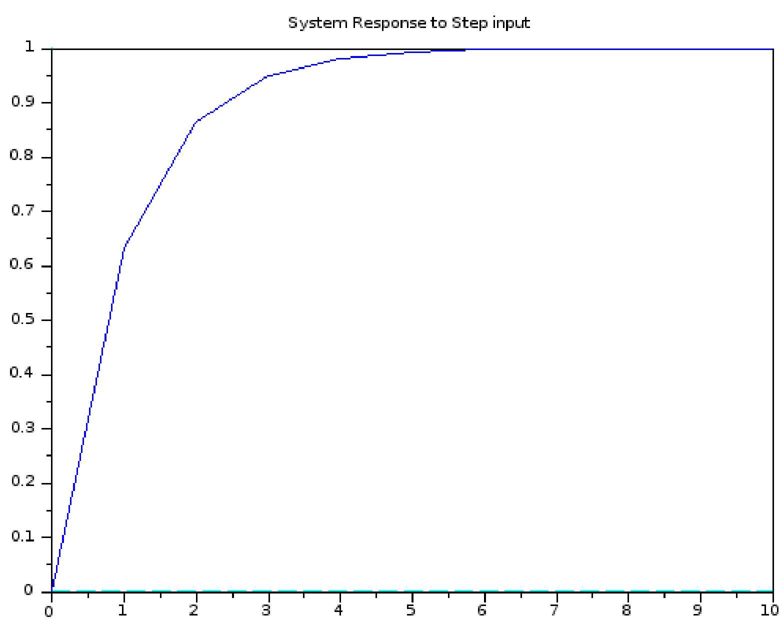

Figure 7.4: Impulse and Step response

```
clc;
clear;
syms s t R L;
Y1=(1/s)-(1/(s+(R/L)));
disp(Y1,'Laplace Transform Of differential Equation is:')
y1=ilaplace(Y1,s,t);
disp(y1,'The Step Response of the System is:');
//Taking R/L=1;
for k=0:10;
    y1(k+1)=1-exp(-k);
end
k=0:10;
plot(k,y1);
title('System Response to Step input');
Y2=(1/(s+(R/L)));
disp(Y2,'Laplace Transform Of differential Equation is:')
y2=ilaplace(Y2,s,t);
disp(y2,'The Impulse Response of the System is:');
for k=0:10;
    y2(k+1)=exp(-k);
end
figure(1);
k=0:10;
plot(k,y2);
title('System Response to impulse input');
```

Scilab code Exa 7.19 Convolution

```
//Scilab Code for Example 7.19 of Signals and
```

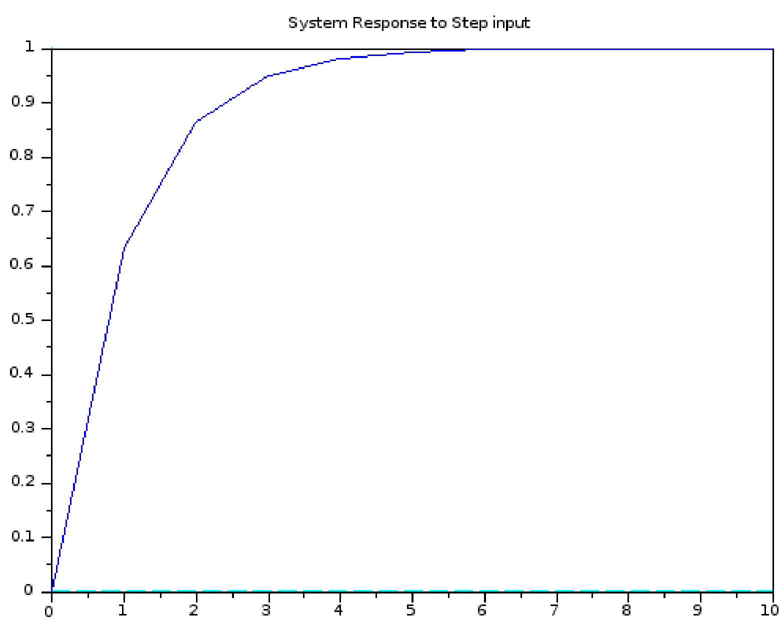

Figure 7.5: Impulse and Step Response

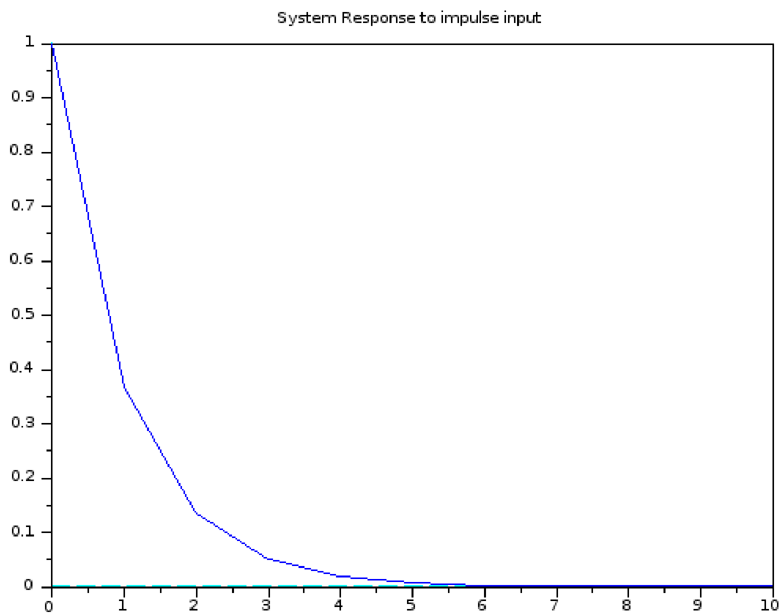

Figure 7.6: Impulse and Step Response

```
    systems by
//P.Ramakrishna Rao
//Convolution of two signals
clc;
clear;
clear x y n;
x=[2,-1,1,0,2];
y=[1,0,-1,2];
n=-1:3;
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot2d3(n,x,-5);
title('x(k)')
xlabel('k')
figure(1);
n=0:3;
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot2d3(n,y,-5);
title('y(k)')
xlabel('k')
z=conv(x,y);
figure(2);
n=-1:6;
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot2d3(n,z,-5);
title('Convoluted signal      z(t)')
xlabel('t')
```

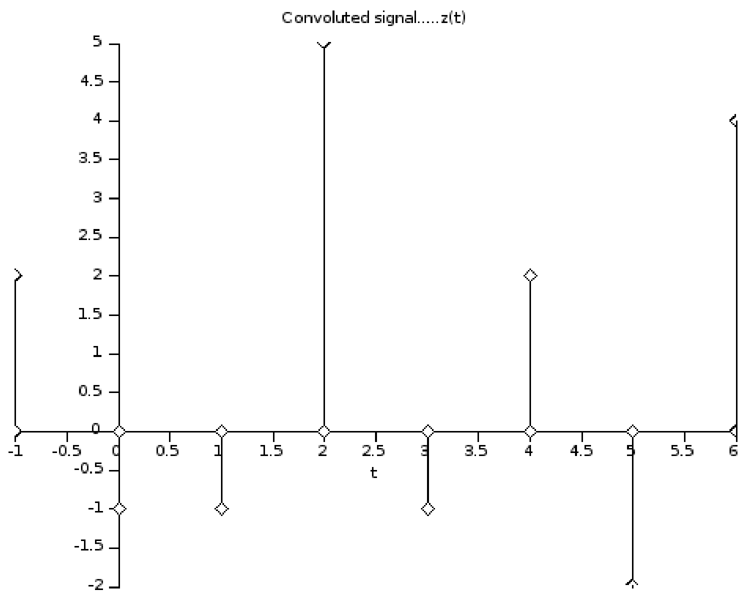

Figure 7.7: Convolution

Scilab code Exa 7.22 Convolution

```
//Scilab Code for Example 7.22 of Signals and
    systems by
//P.Ramakrishna Rao
//Convolution of two signals
clc;
clear;
clear x y n;
for n=0:10;
    x(n+1)=(3/4)^n*u(n);
end
c = gca();
c.y_location = "origin";
c.x_location = "origin";
n=0:10;
plot2d3(n,x,-4);
title('x(n)')
xlabel('n')
for n=0:10;
    y(n+1)=u(n);
end
figure(1);
n=0:10;
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot2d3(n,y,-4);
title('y(k)')
xlabel('k')
z=conv(x,y);
figure(2);
n=0:20;
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot2d3(n,z,-4);
title('Convoluted signal     w(t)');
```

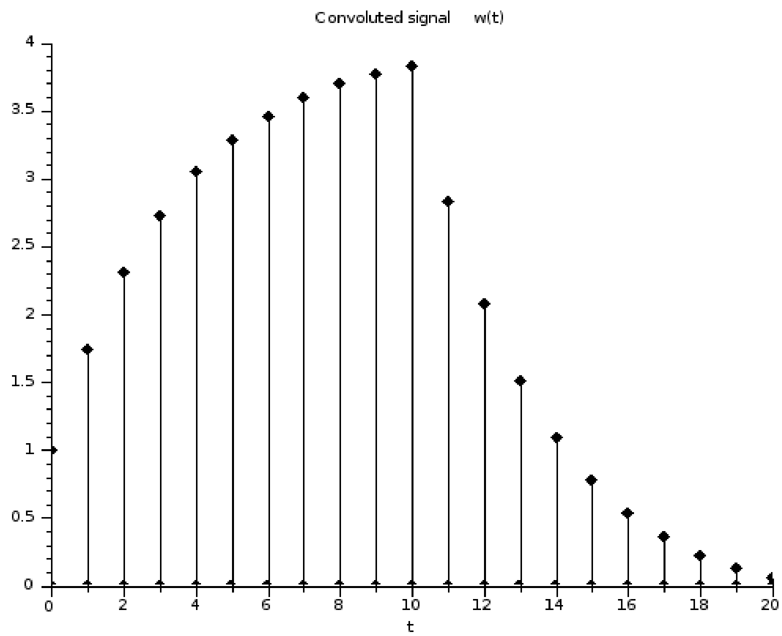

Figure 7.8: Convolution

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

Scilab code Exa 7.23 Convolution

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

```
clear;
clear x y n;
for n=0:10;
    x(n+1)=(0.5)^n*u(n-2);
end
c = gca();
c.y_location = "origin";
c.x_location = "origin";
n=0:10;
plot2d3(n,x,-4);
title('x(n)')
xlabel('n')
for n=0:10;
    y(n+1)=u(n);
end
figure(1);
n=0:10;
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot2d3(n,y,-4);
title('y(k)')
xlabel('k')
z=conv(x,y);
figure(2);
n=0:20;
c = gca();
c.y_location = "origin";
c.x_location = "origin";
plot2d3(n,z,-4);
title('Convoluted signal      z(t)')
xlabel('t')
```

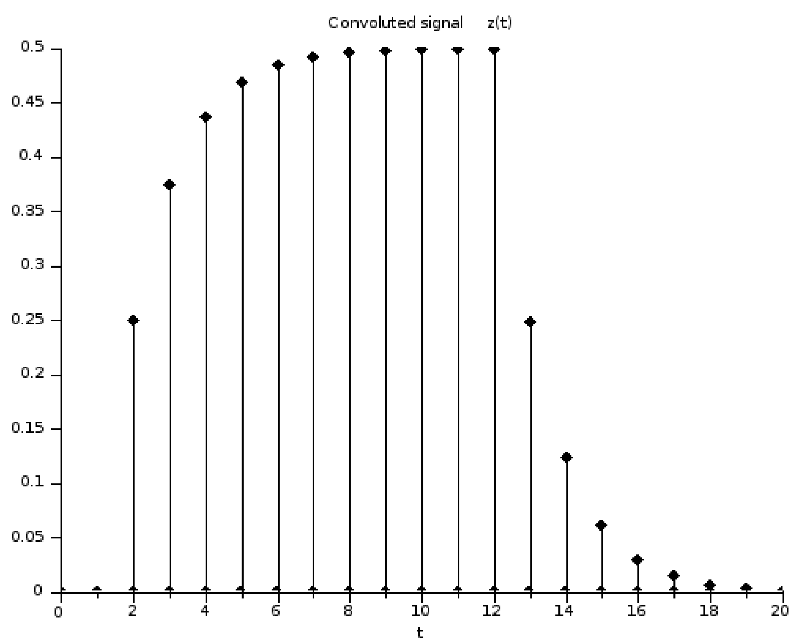

Figure 7.9: Convolution

Scilab code Exa 7.24 Step Response

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

Scilab code Exa 7.25 Step response

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

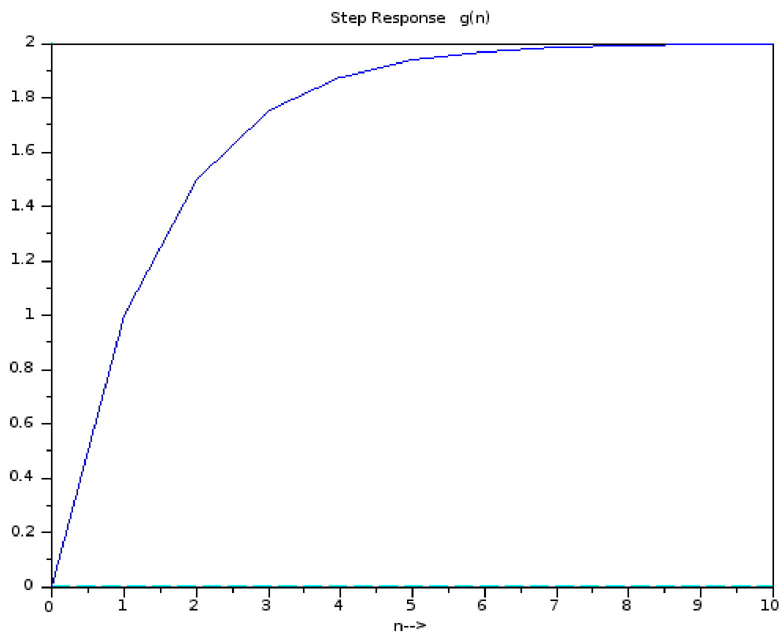

Figure 7.10: Step Response

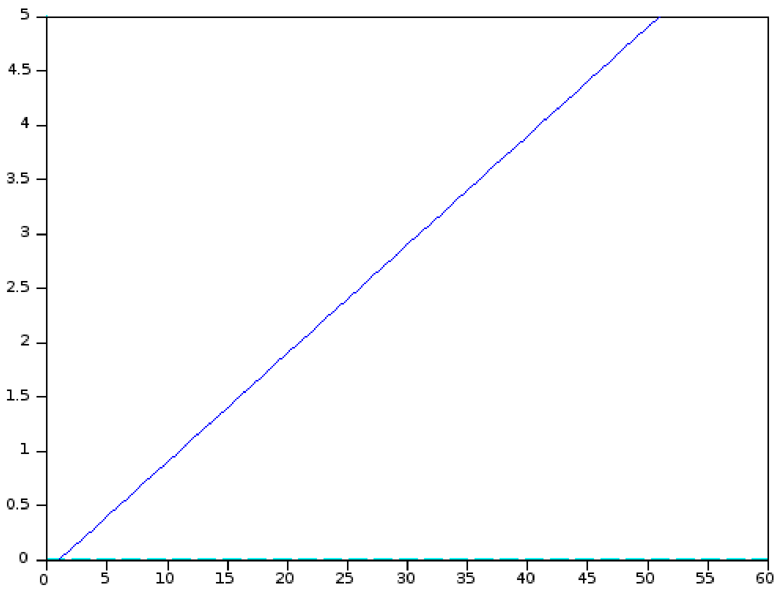

Figure 7.11: Step response

```
    ');
y2=2*(a^n-1)/(a-1);
disp(y2,'The Step Response of the System is:');
q=1;
a=0.5;
for k=0:0.1:5;
y2(q)=2*((a^k-1)/(a-1));
q=q+1;
end
k=0:0.1:5;
plot(k,y2);
```

Scilab code Exa 7.33 Magnitude and Phase Response

```
//Scilab Code for Example 7.33 of Signals and
   systems by
//P.Ramakrishna Rao
//Plotting the magnitude and phase responses
clc;
clear;
T=1;
n=1;
for w=0:0.1:20;
    hmag(n)=2*sin(w*T/2);
    n=n+1;
end
n=1;
for w=0:0.1:20;
    hphase(n)=%pi/2-(w*T/2);
    n=n+1;
end
//Magnitude plot
w=0:0.1:20;
plot(w,hmag);
title('Magnitude Plot');
xlabel('w');
ylabel('|H(e^jw)|');
figure(1);
//Phase Plot
w=0:0.1:20;
plot(w,hphase);
title('Phase Plot');
xlabel('w');
ylabel('theta(wT)');
```

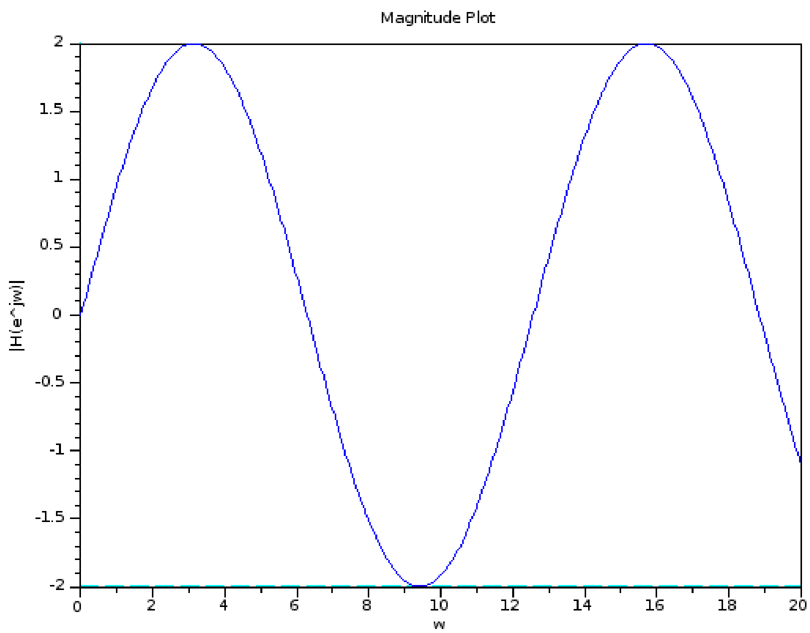

Figure 7.12: Magnitude and Phase Response

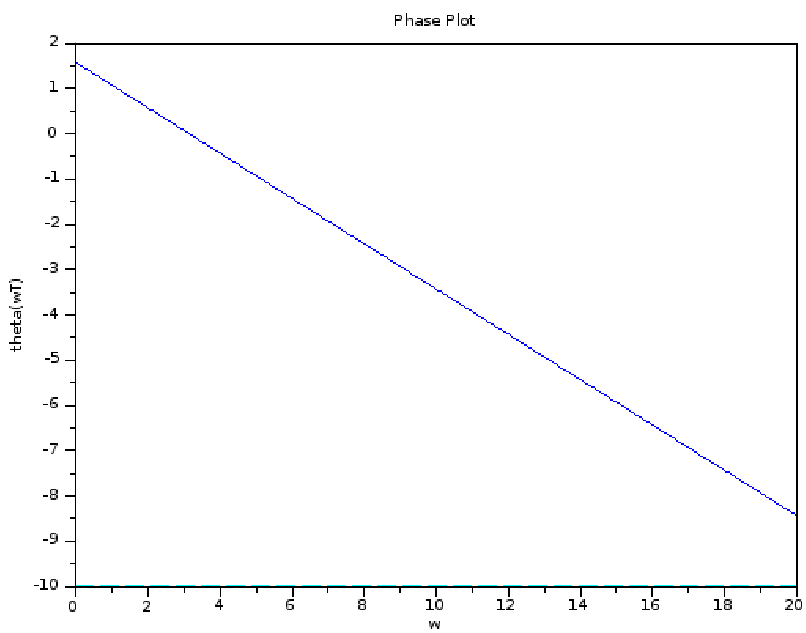

Figure 7.13: Magnitude and Phase Response

Chapter 8

Systems Time and Frequency Domain Analysis

Scilab code Exa 8.1 Magnitude and Phase Responses

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

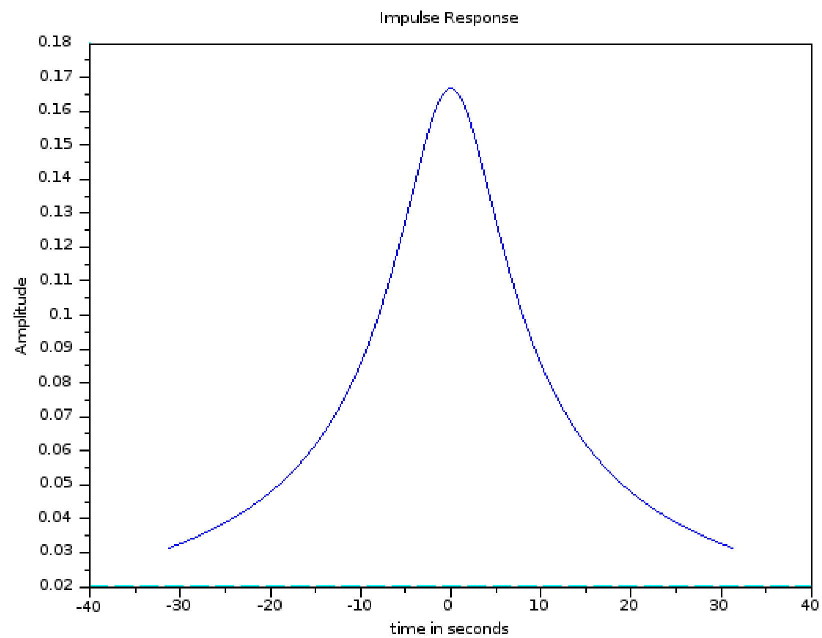

Figure 8.1: Magnitude and Phase Responses

```
title('Impulse Response');
ylabel('Amplitude');
xlabel('time in seconds');
figure(1);
w=-5*2*%pi:0.01:5*2*%pi;
plot(w,Hphs);
title('Phase Response');
ylabel('theta(w)');
xlabel('w in radians/second');
```

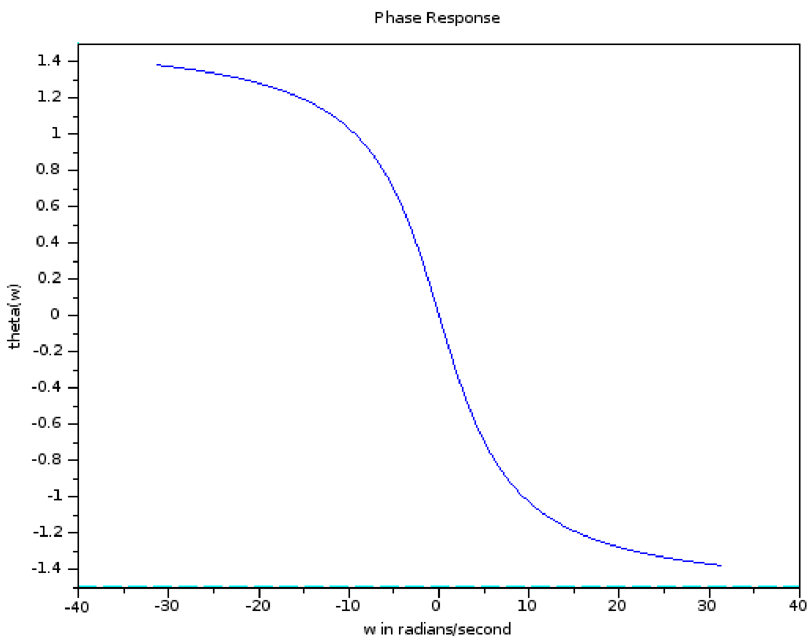

Figure 8.2: Magnitude and Phase Responses

Scilab code Exa 8.2 Transfer Function of system

```
//Scilab Code for Example 8.2 of Signals and systems
   by
//P.Ramakrishna Rao
//Second order system
clear;
clc;
syms t;
L=1;
R=1.2;
C=10^-6;
Rcr=2*sqrt(L/C);
eta=R/Rcr;
disp(eta,'Damping Ratio=',Rcr,'Critical Resistance (
  ohm)','(i)');
Wn=1/sqrt(L*C);
disp(Wn,'Undamped Natural Frequency(Hz)','(ii)');
n=1;
h=(1250*sin(800*t))*exp(-600*t);
H=laplace(h,t,'jW');
for t=0:0.1:2;
    h(n)=(1250*sin(800*t))*exp(-600*t);//Impulse
       Response
    n=n+1;
end
t=0:0.1:2;
plot(t,h);
title('Impulse Response');
ylabel('Amplitude');
xlabel('time in seconds');
disp(H,'(iv)Transfer Function(H(jw)):');
```

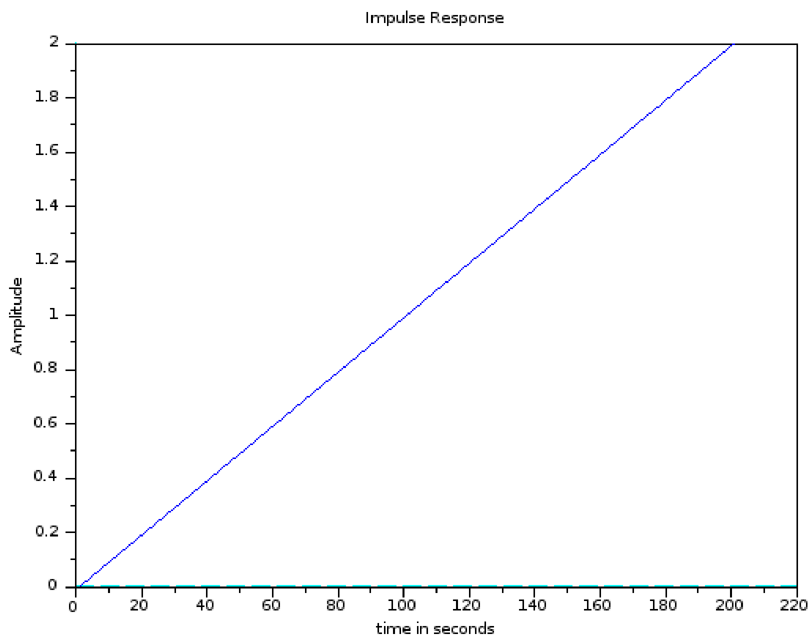

Figure 8.3: Transfer Function of system

Scilab code Exa 8.3 Magnitude and Phase Responses

```
//Scilab Code for Example 8.3 of Signals and systems
    by
//P.Ramakrishna Rao
clear;
clc;
syms n w;
h=(0.3)^n;
disp(h,'The impulse response of the system is:');
H=1/(1-(0.3*exp(-%i*w)));
disp(H,'The Transfer function of the system is:');
n=1;
for w=-%pi:0.1:%pi
    Hmag(n)=1/sqrt(1.09-(0.6*cos(w)));
    Hphs(n)=-atan(0.3*sin(w)/(1-0.3*cos(w)));
    n=n+1;
end
w=-%pi:0.1:%pi;
c = gca();
c.y_location = "origin";
c.x_location = "origin";
c.thickness=2;
plot(w,Hmag);
title('Magnitude Sketch');
ylabel('Amplitude');
xlabel('W in radians');
figure(1);
w=-%pi:0.1:%pi;
c = gca();
c.y_location = "origin";
c.x_location = "origin";
c.thickness=2;
plot(w,Hphs);
title('Phase Response');
ylabel('theta(w)');
xlabel('W in radians')
```

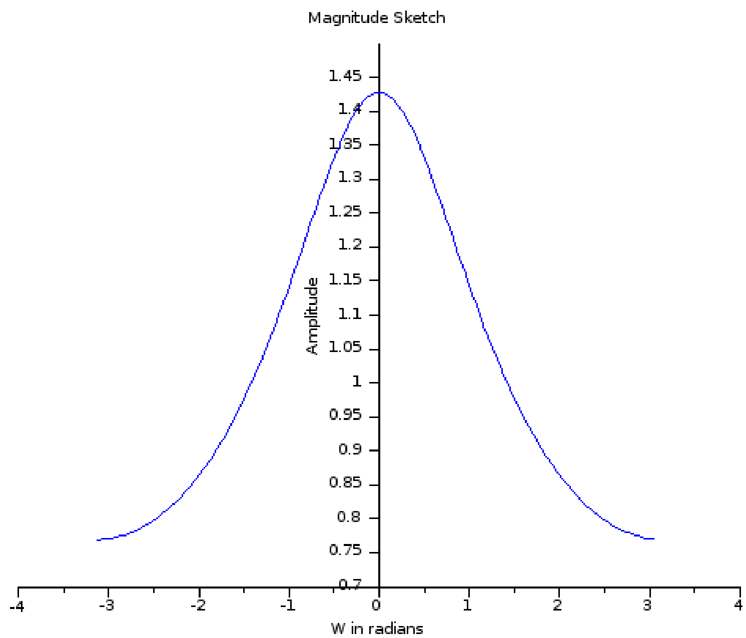

Figure 8.4: Magnitude and Phase Responses

Scilab code Exa 8.4 system response

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

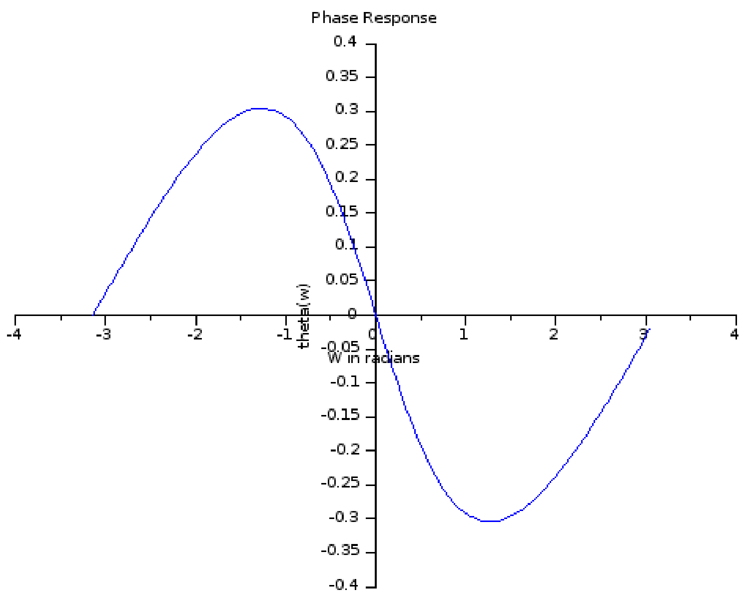

Figure 8.5: Magnitude and Phase Responses

```
z = %z;
syms n z1;
X  =(z-1)/((z+(1/2))*(z-(1/5)))
disp(X,'(i) System Function is:');
X1 = denom(X);
zp = roots(X1);
X1 = (z1-1)/((z1+(1/2))*(z1-(1/5)));
F1 = X1*(z1^(n-1))*(z1-zp(1))
F2 = X1*(z1^(n-1))*(z1-zp(2))
h1 = limit(F1,z1,zp(1))
h2 = limit(F2,z1,zp(2))
h = h1+h2;
disp('(iv) Unit sample response sequence:')
disp(h*"u(n)",'h[n]=');
for n=0:10;
    w(n+1)=(- 5.714*(0.2)^n- 4.285*(- 0.5^n));
end
n=0:10;
plot(n,w);
title('(iii)    Unit Sample Response Sequence');
disp('where z=e^jw ',X,'(i) Transfer Function is:');
```

Scilab code Exa 8.5 System function

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

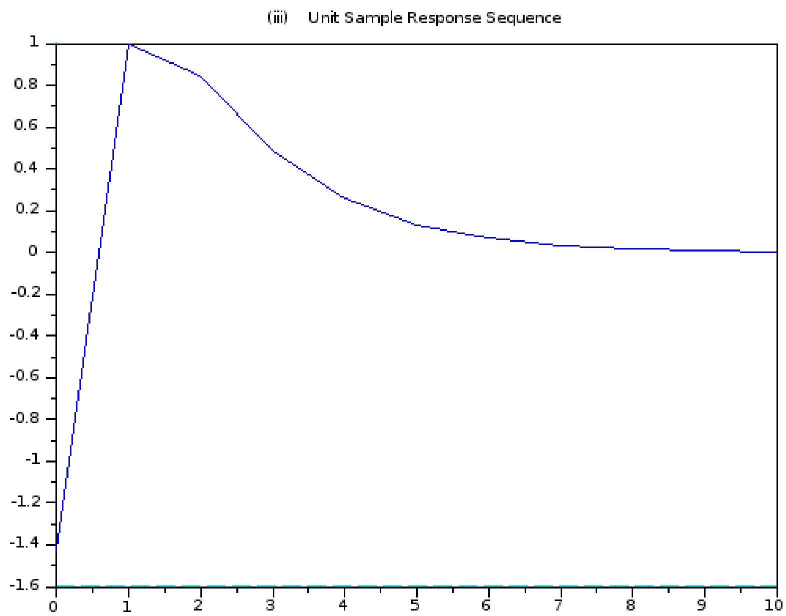

Figure 8.6: system response

```
disp(X,'(i) System Function is:');
X1 = denom(X);
zp = roots(X1);
X1 = z1^2/(z1-0.5)^2;
F1 = X1*(z1^(n-1))*(z1-0.5)^2;
h1 = limit(F1,z1,zp(1));
h = h1;
disp('(iii) Unit step response sequence:')
disp(h*"u(n)",'h[n]=');
for n=0:20;
    w1(n+1)=1-(0.5)^n;
end
n=0:20;
plot(n,w1);
title('(ii)    Unit Sample Response Sequence');
disp('where z=e^jw ',X,'(iv) Transfer Function (H(e^jw))is:');
```

Scilab code Exa 8.8 Transfer function

```
//Scilab Code for Example 8.8 of Signals and systems by
//P.Ramakrishna Rao
clear;
clc;
syms R L C s I;
X=I*(R+1/(C*s));
Y=R*I;
Z=Y/X;
disp(Z,'(a) RC High pass Filter: H(s)');
X=I*(L*s+1/(C*s));
Y=I/(C*s);
Z=Y/X;
```

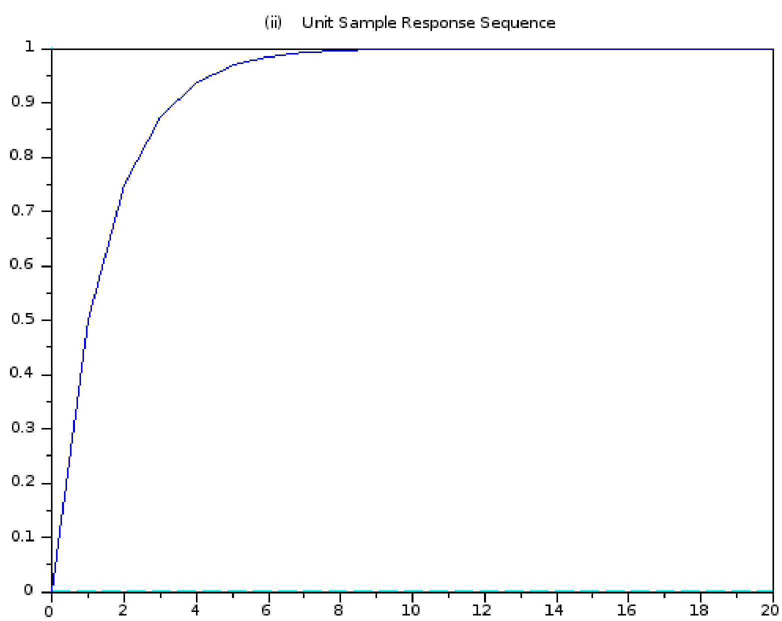

Figure 8.7: System function

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

Scilab code Exa 8.9 Transfer function by block diagram reduction

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

Scilab code Exa 8.10 Realization Structure

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

```
H=s^3/(s^3+6*s^2++11*s+6);
H1=denom(H);
a=roots(H1);
H2=5*(s1^3/(s1-a(1)))*(s1/(s1-a(2)))*(s1/(s1-a(3)));
disp(H2,'(ii) Cascade Form: H(s)');
h1=s1^3*(s1-a(1))/((s1+1)*(s1+2)*(s1+3));
h2=s1^3*(s1-a(2))/((s1+1)*(s1+2)*(s1+3));
h3=s1^3*(s1-a(3))/((s1+1)*(s1+2)*(s1+3));
A = limit(h1,s1,a(1))/(s-a(1));
B = limit(h2,s1,a(2))/(s-a(2));
C = limit(h3,s1,a(3))/(s-a(3));
Z=5*(1+A+B+C);
disp(Z,'(iii)Parallel Form Realization: H(s)');
```

Scilab code Exa 8.12 Transfer function Realization

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

```
w2=limit(F2,z,a(2))
w=w1+w2;
disp(w,'(b) h(n))=');
z=%z;
x11=z*x1/3
x12=z*x2/3
disp(x11,"(c) Parallel realization: H1(z)");
disp(x12," H2(z)")
```

Scilab code Exa 8.13 Parallel form realization

```
//Scilab Code for Example 8.13 of Signals and systems by
//P.Ramakrishna Rao
clear;
clc;
syms z1;
z=%z;
H=poly(0,'z');
H=z*(z+0.5)*(z+0.25)/((z-0.5)*(z-0.25)*(z-0.125));
H1=denom(H);
a=roots(H1)
h1=(z1+0.5)*(z1+0.25)/((z1-0.5)*(z1-0.25)*(z1-0.125))*(z1-a(1));
h2=(z1+0.5)*(z1+0.25)/((z1-0.5)*(z1-0.25)*(z1-0.125))*(z1-a(2));
h3=(z1+0.5)*(z1+0.25)/((z1-0.5)*(z1-0.25)*(z1-0.125))*(z1-a(3));
A=z*limit(h1,z1,a(1))/(z-a(1));
B=z*limit(h2,z1,a(2))/(z-a(2));
C=z*limit(h3,z1,a(3))/(z-a(3));
Z=A+B+C;
disp(Z,'Parallel Form Realisation: H(z)');
```

Chapter 9

Convolution and Correlation Continuous Time Signals

Scilab code Exa 9.1 Convolution

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

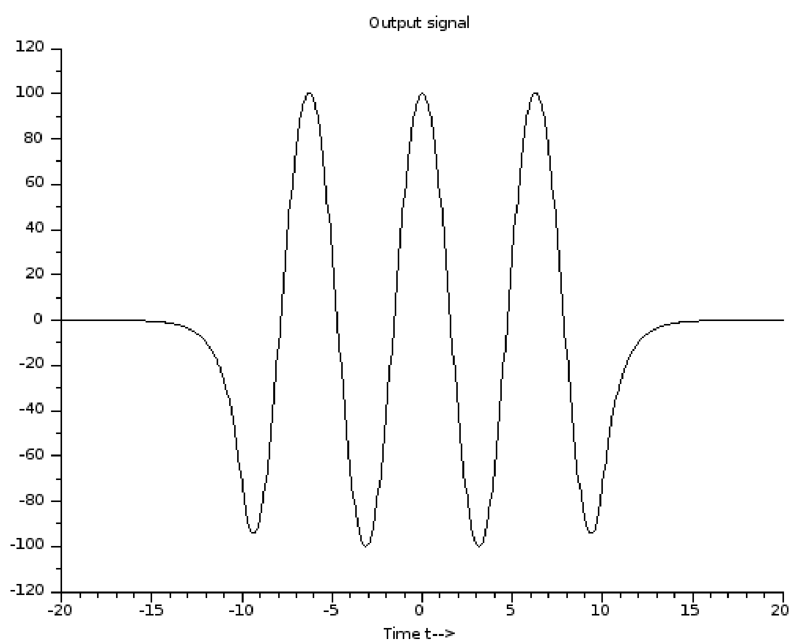

Figure 9.1: Convolution

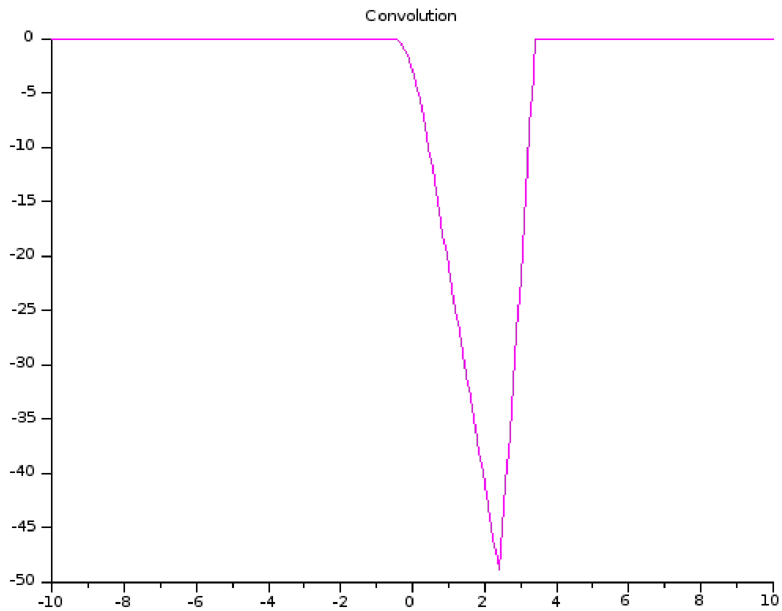

Figure 9.2: Convolution

Scilab code Exa 9.3 Convolution

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

```
    x(n)=-2*u(t-0.5)+2*u(t-1.5);
    y(n)=(t+1)*u(t+1)-(t+1)*u(t-2);
    n=n+1;
end
z=conv(x,y);
t=-5:0.1:5;
plot2d2(t,x,5);
title('x(t)');
figure(1);
t=-5:0.1:5;
plot2d(t,y,2);
title('y(t)');
figure(2);
t=-10:0.1:10;
plot2d(t,z,6);
title('Convolution');
```

Scilab code Exa 9.4 Autocorrelation

```
//Scilab Code for Example 9.4 of Signals and systems
    by
//P.Ramakrishna Rao
clc;
clear;
T=1;
n=1;
for t=-T/2:0.01:T/2;
    x(n)=10*cos(%pi*t/T);
    n=n+1;
end
t=-T/2:0.01:T/2;
plot(t,x);
title('x(t)');
```

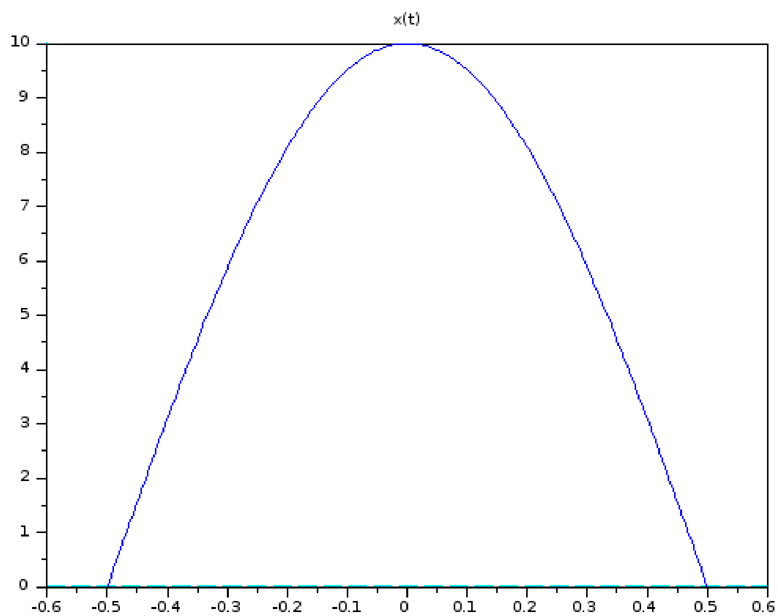

Figure 9.3: Autocorrelation

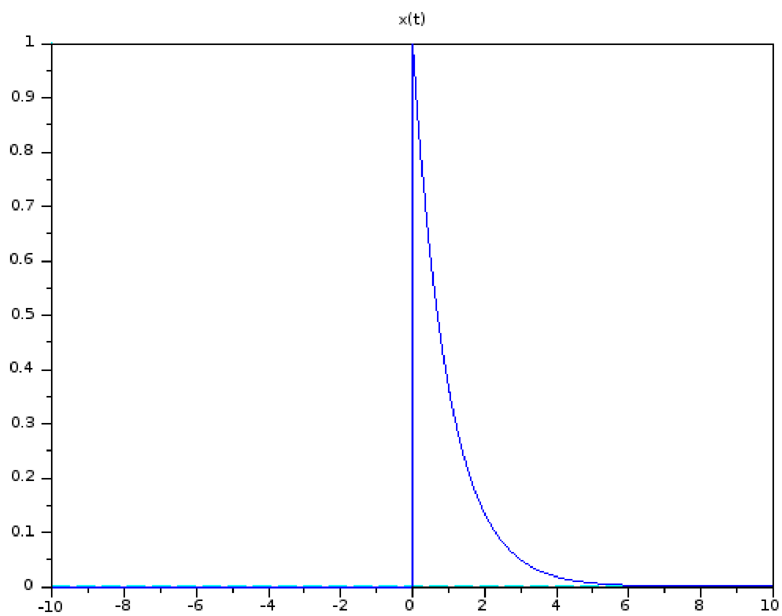

Figure 9.4: Autocorrelation

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

Scilab code Exa 9.5 Autocorrelation

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

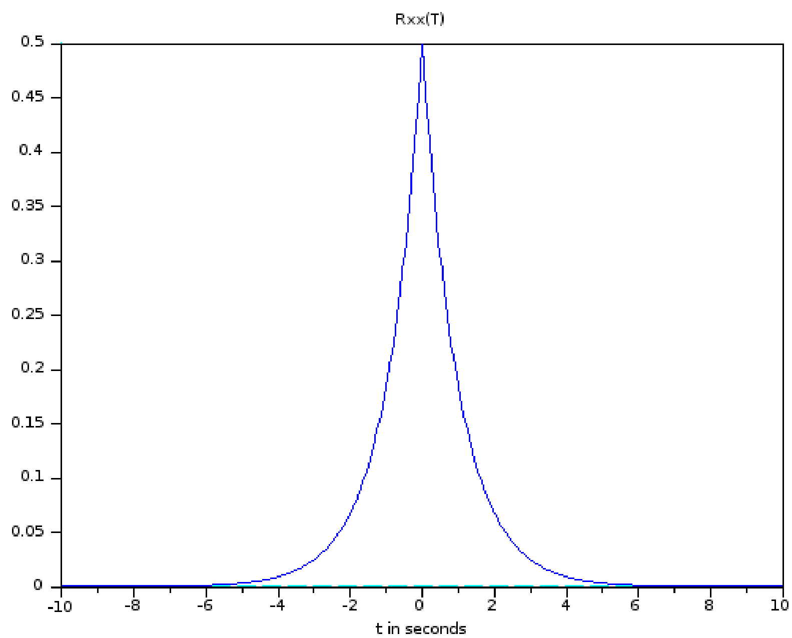

Figure 9.5: Autocorrelation

```
clc;
clear;
clear z x y t T;
n=1;
for t=-10:0.01:10;
    x(n)=exp(-t)*u(t);
    n=n+1;
end
t=-10:0.01:10;
plot(t,x);
title('x(t)');
n=1;
for T=-10:0.1:0;
    z(n)=integrate('exp(-t)*exp(T-t)','t',0,1000);
    n=n+1;
end
n=1;
for T=0.1:0.1:10;
    z(n+101)=integrate('exp(-t)*exp(T-t)','t',T
       ,1000);
    n=n+1;
end
figure(1);
T=-10:0.1:10;
plot(T,z);
title('Rxx(T)');
xlabel('t in seconds');
```

Scilab code Exa 9.6 Autocorrelation

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

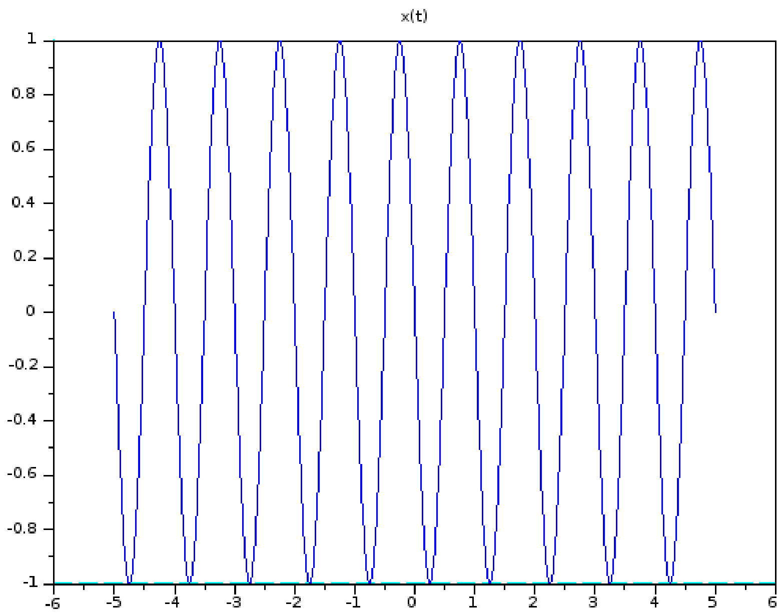

Figure 9.6: Autocorrelation

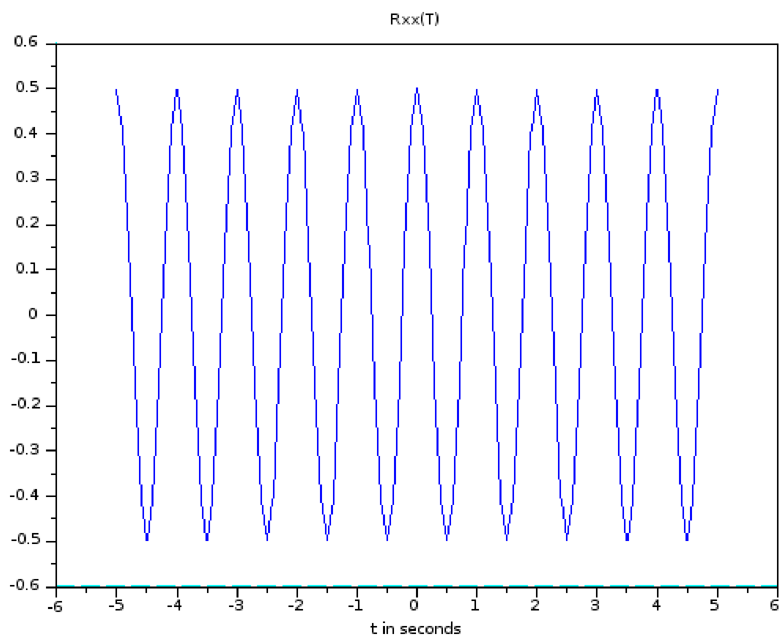

Figure 9.7: Autocorrelation

```
    by
//P.Ramakrishna Rao
clc;
clear;
clear z x y t T;
A=1;
n=1;
wo=2*%pi;
theta=%pi/2;
for t=-5:0.01:5;
    x(n)=A*cos(wo*t+theta);
    n=n+1;
end
t=-5:0.01:5;
plot(t,x);
title('x(t)');
n=1;
for T=-5:0.1:5;
    z(n)=(A^2/2)*cos(wo*T);
    n=n+1;
end
figure(1);
T=-5:0.1:5;
plot(T,z);
title('Rxx(T)');
xlabel('t in seconds')
```

Scilab code Exa 9.7 Autocorrelation

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

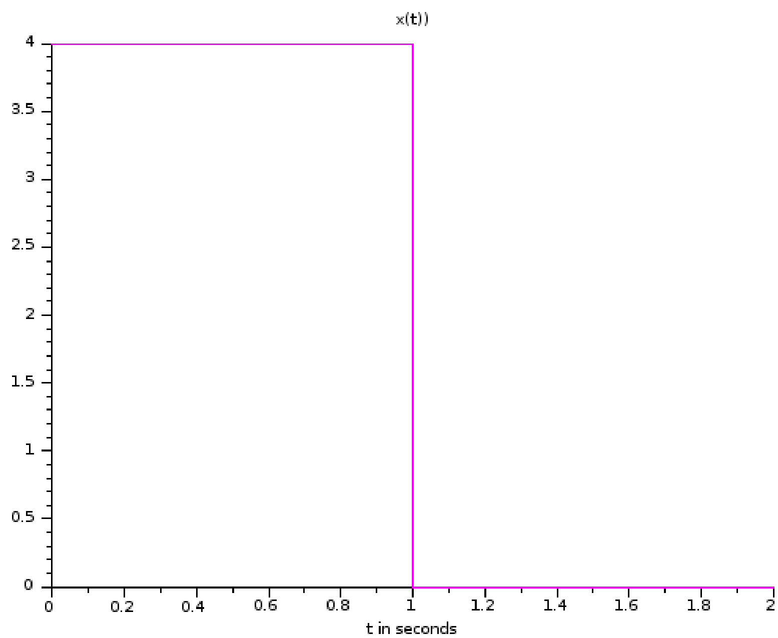

Figure 9.8: Autocorrelation

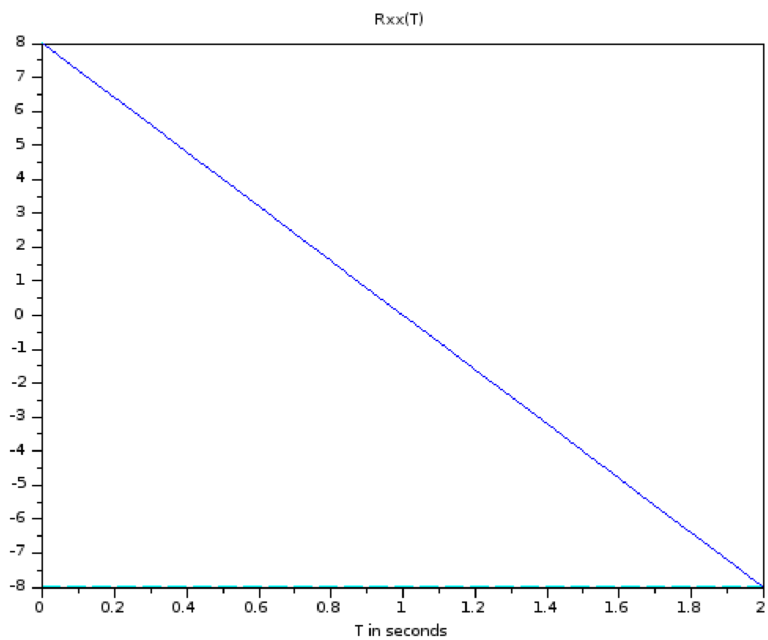

Figure 9.9: Autocorrelation

```
//P.Ramakrishna Rao
clc;
clear;
clear z x y t T A;
A=4;
T0=2;
n=1;
for T=0:0.1:T0;
    if T<T0/2 then
        x(n)=A;
    else x(n)=0;
        end
    n=n+1;
end
T=0:0.1:T0;
plot2d2(T,x,6);
title('x(t))');
xlabel('t in seconds');
n=1;
for T=0:0.1:T0;
    z(n)=(1/T0)*integrate('A^2','t',T,T0/2);
    n=n+1;
end
T=0:0.1:T0;
figure(1);
plot(T,z);
title('Rxx(T)');
xlabel('T in seconds');
```

Scilab code Exa 9.10 Autocorrelation Function

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

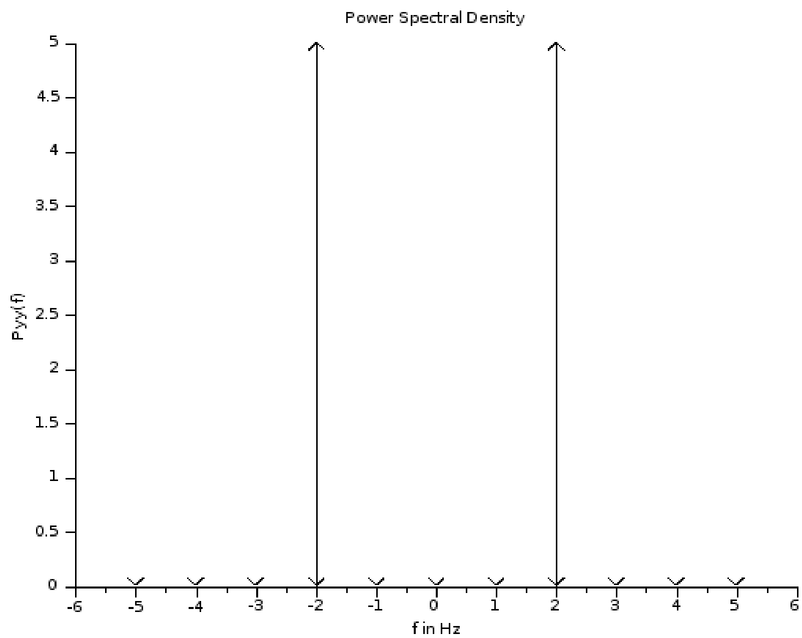

Figure 9.10: PSD

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

Scilab code Exa 9.11 PSD

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

```
//Power Spectral Density
clear;
clc;
clear f n;
fo=2;
n=1;
RC=1/(2*%pi*10^3);
w=4*%pi*10^3
H=1/(1+%i*w*RC);
disp(H,'H(f)');
X=(abs(H))^2;
for f=-5:5
    Pxx(n)=25*[delta(f-fo)+delta(f+fo)];
    Pyy(n)=X*Pxx(n);
    n=n+1;
end
disp(Pxx,'Pxx=');
disp(Pyy,'Pyy=');
f=-5:5;
plot2d3(f,Pyy,-2);
title('Power Spectral Density');
ylabel('Pyy(f)');
xlabel('f in Hz');
```

Chapter 10

Discrete Time Convolution and Correlation

Scilab code Exa 10.1 Cross Correlation

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

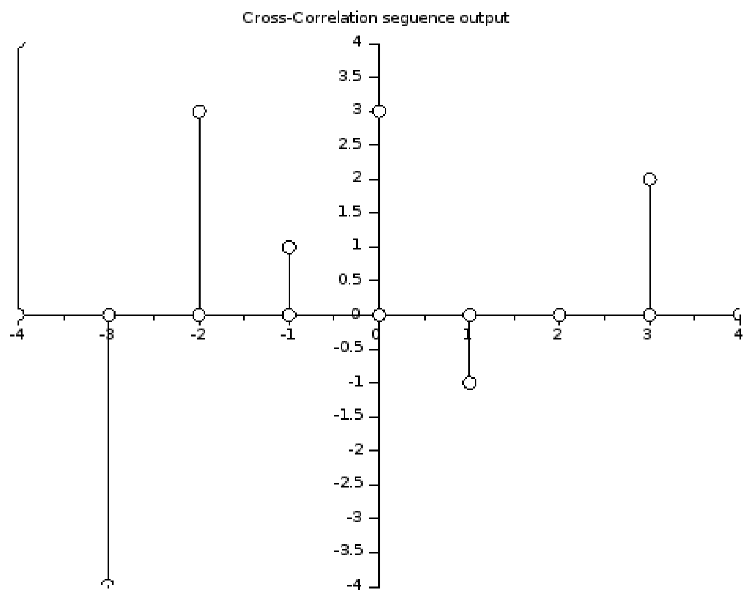

Figure 10.1: Cross Correlation

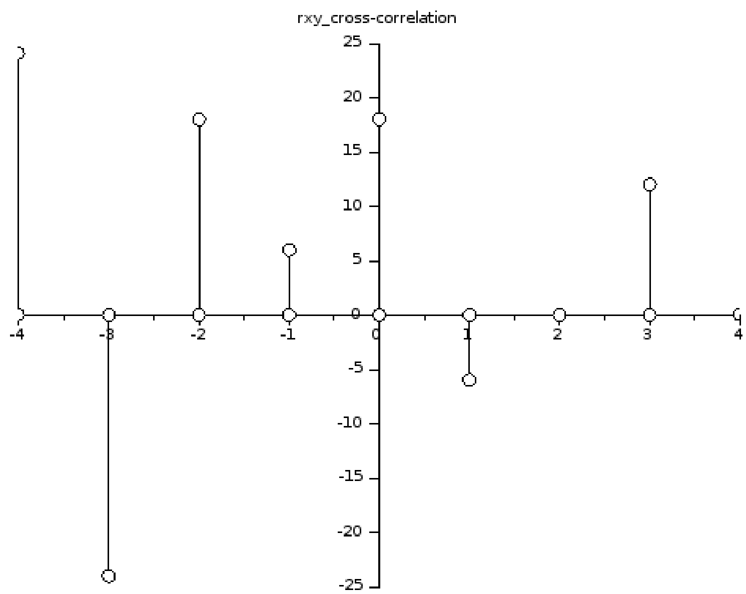

Figure 10.2: Cross Correlation Sequence

```
title('Cross-Correlation seguence output');
```

Scilab code Exa 10.2 Cross Correlation Sequence

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

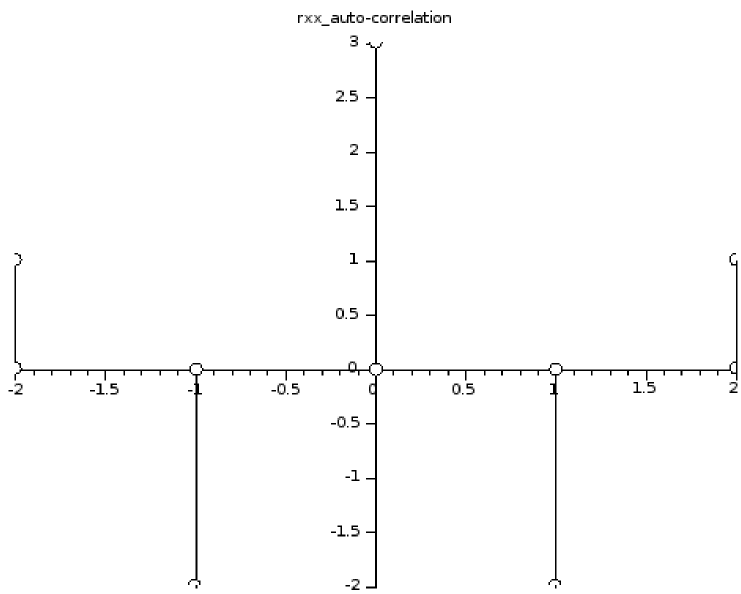

Figure 10.3: Auto Correlation

```
y=[0,3,0,-3,6];
//computation of cross correlation sequence;
n1 = max(size(y))-1;
n2 = max(size(x))-1;
r = xcorr(x,y,n1);
n=-4:4;
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d3(n,r,-9);
title('rxy_cross-correlation');
```

Scilab code Exa 10.4 Auto Correlation

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

Scilab code Exa 10.5 Auto Correlation

```
//Scilab Code for Example 10.5 of Signals and systems by
//P.Ramakrishna Rao
//Auto Correlation
clear;
clc;
clear x n a;
k=1;
a=0.8;
for n=-30:30;
    x(k)=a^(-n)*u(-n);
    k=k+1;
end
```

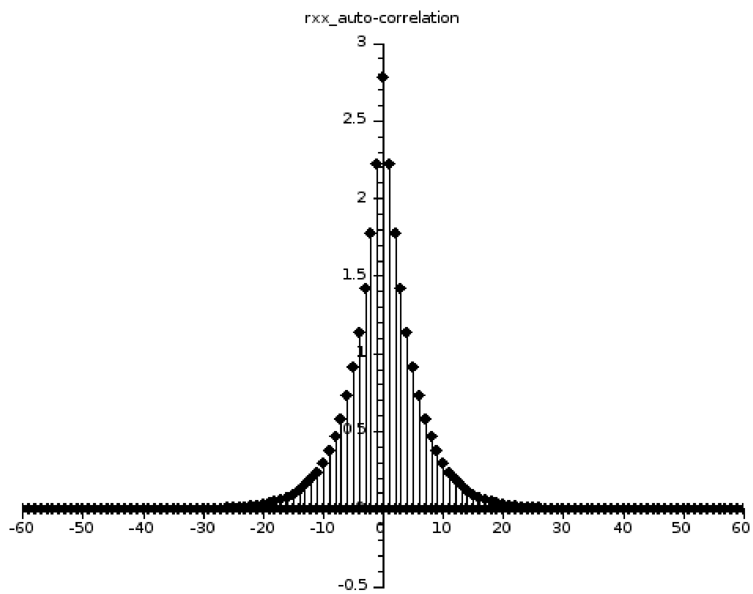

Figure 10.4: Auto Correlation

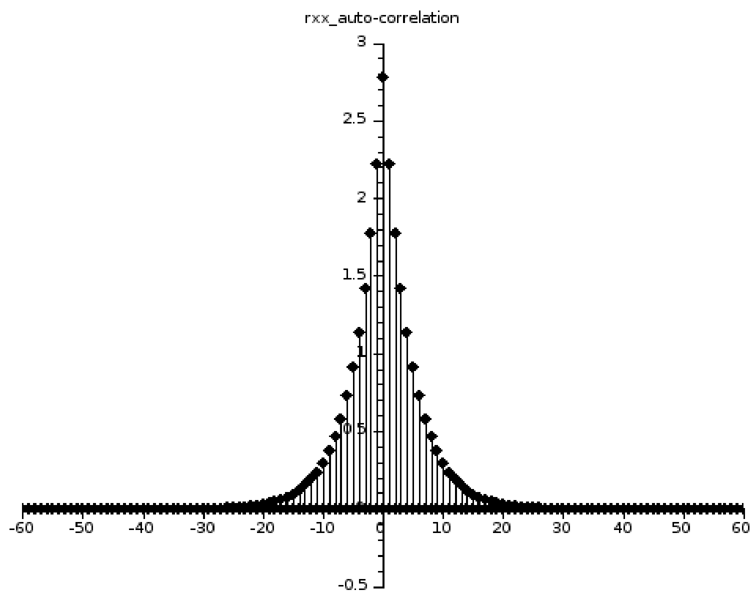

Figure 10.5: Auto Correlation

```
length(x)
//computation of auto correlation sequence;
r = xcorr(x);
n=-60:60;
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d3(n,r,-4);
title('rxx_auto-correlation');
```

Scilab code Exa 10.8 Auto Correlation

```
//Scilab Code for Example 10.8 of Signals and systems by
//P.Ramakrishna Rao
//Auto Correlation
clear;
clc;
clear x n a;
k=1;
a=0.8;
for n=-30:30;
    x(k)=a^(n)*u(n);
    k=k+1;
end
length(x)
//computation of auto correlation sequence;
r = xcorr(x);
n=-60:60;
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d3(n,r,-4);
title('rxx_auto-correlation');
```

Scilab code Exa 10.11 System response

```
//Scilab Code for Example 10.11 of Signals and systems by
//P.Ramakrishna Rao
//System Identification
//Cross Correlation
clear;
clc;
clear x n a y h z;
```

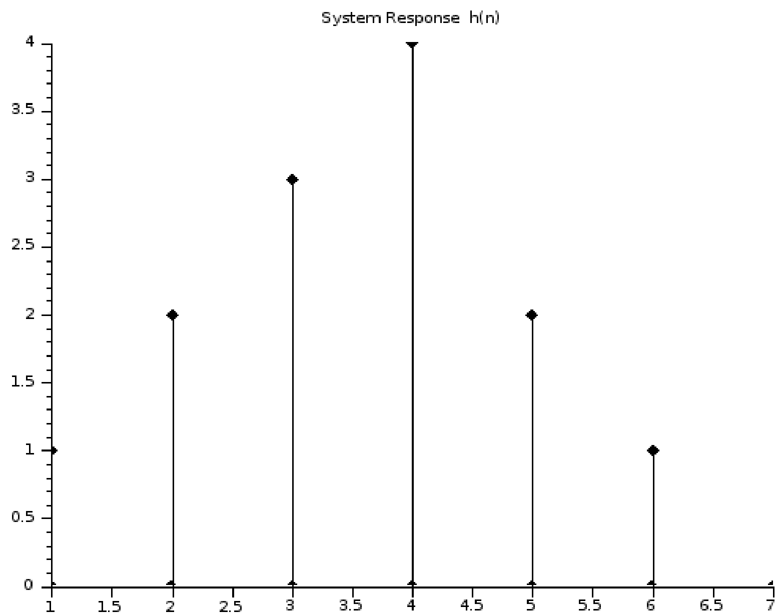

Figure 10.6: System response

```
x=[1,1,1,1,0,0];
y=[1,3,6,10,12,12,10,6,3,1];
h(1)=y(1)/x(1);
for n=2:7
    for k=1:n-1
        z(k)=(h(k)*x(n-k));
    h(n)=(y(n)-sum(z));
    end
end
disp(h,'h(n)');
n=1:7;
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d3(n,h,-4);
title('System Response  h(n)');
```

Chapter 11

Hilbert Transform Continuous and Discrete

Scilab code Exa 11.1 Hilbert Transform

```
//Scilab Code for Example 11.1 of Signals and systems by
//P.Ramakrishna Rao
//Hilbert Transform
clc;
clear;
n=1;
for t=-1:0.01:1
    xr(n)=sin(2*%pi*t);
    n=n+1;
end
//Computing Hilbertb Transform
x=hilbert(xr);
t=-1:0.01:1;
plot(t,xr);
```

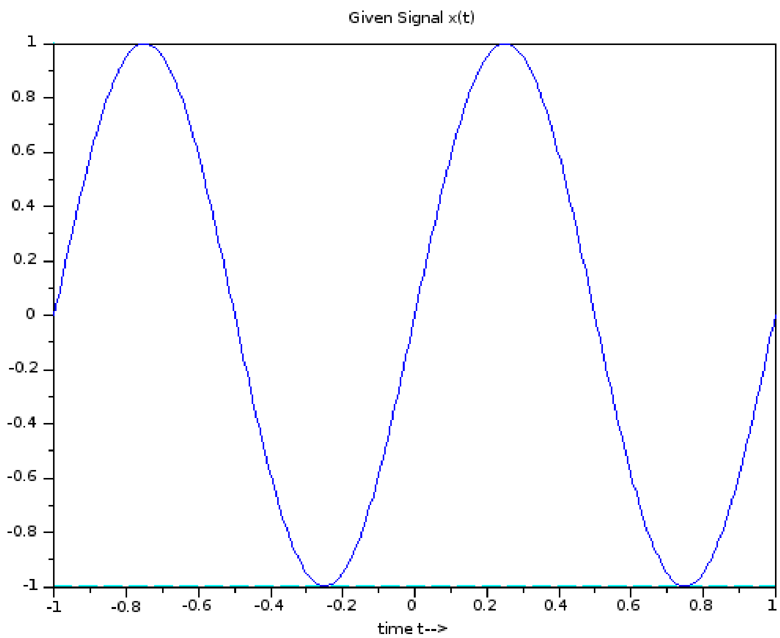

Figure 11.1: Hilbert Transform

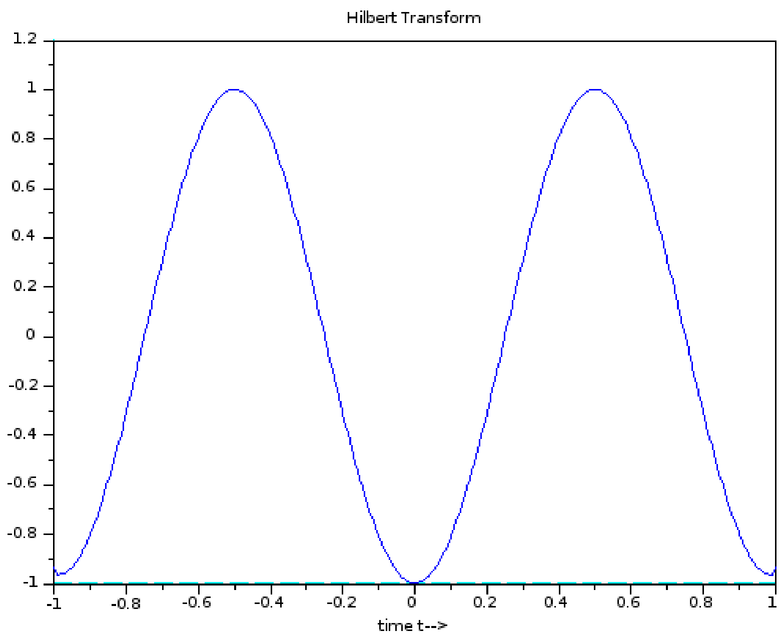

Figure 11.2: Hilbert Transform

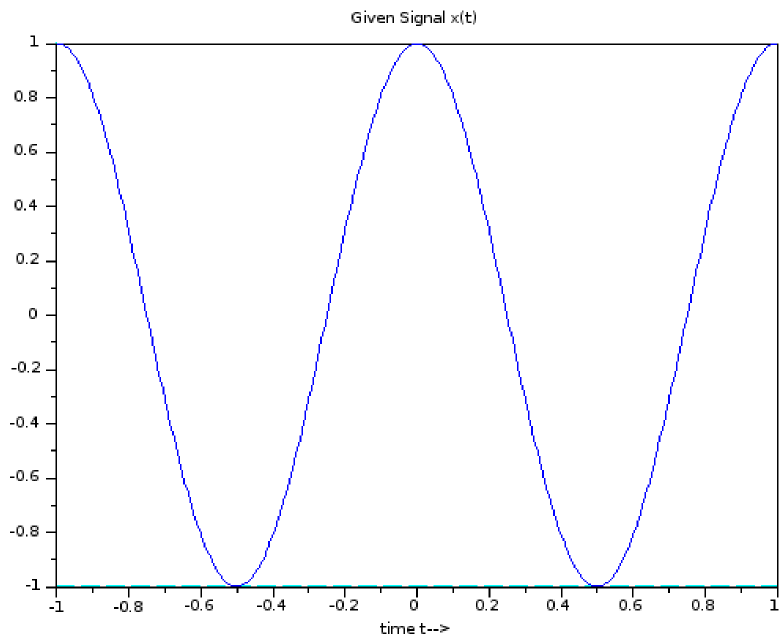

Figure 11.3: Hilbert Transform

```
title('Given Signal x(t)');
xlabel('time t-->');
figure(1);
t=-1:0.1:1;
plot(t,imag(x));
title('Hilbert Transform');
xlabel('time t-->');
```

Scilab code Exa 11.2 Hilbert Transform

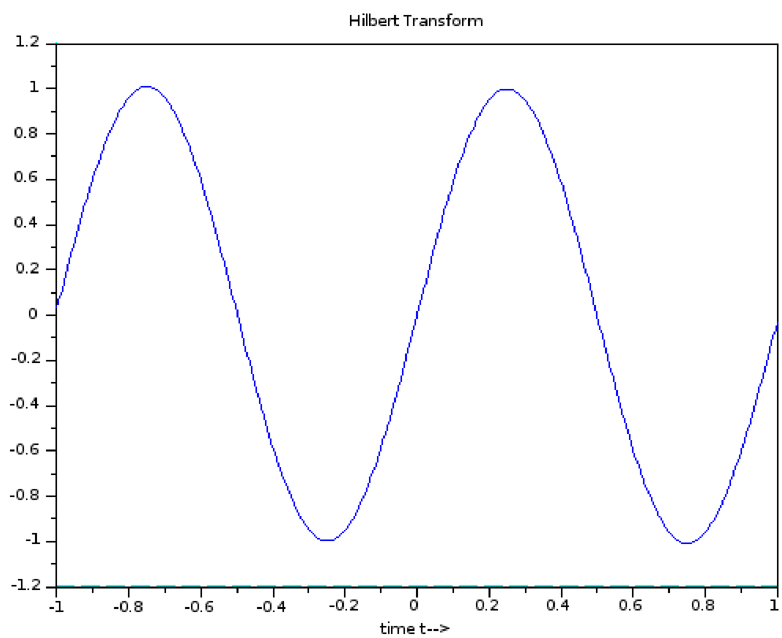

Figure 11.4: Hilbert Transform

```
//Scilab Code for Example 11.2 of Signals and systems by
//P.Ramakrishna Rao
//Hilbert Transform
clc;
clear;
n=1;
for t=-1:0.01:1
    xr(n)=cos(2*%pi*t);
    n=n+1;
end
//Computing Hilbertb Transform
x=hilbert(xr);
t=-1:0.01:1;
plot(t,xr);
title('Given Signal x(t)');
xlabel('time t-->');
figure(1);
t=-1:0.01:1;
plot(t,imag(x));
title('Hilbert Transform');
xlabel('time t-->');
```

Scilab code Exa 11.3 Hilbert Transform

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

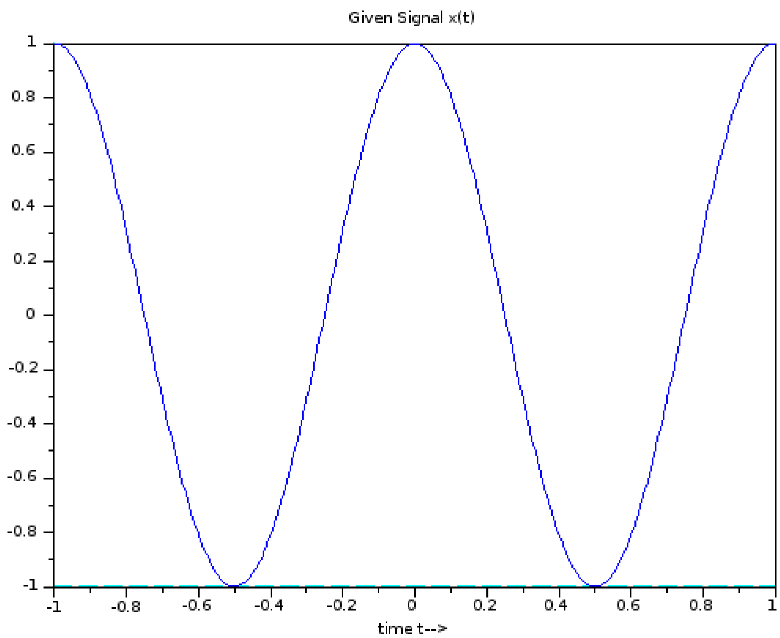

Figure 11.5: Hilbert Transform

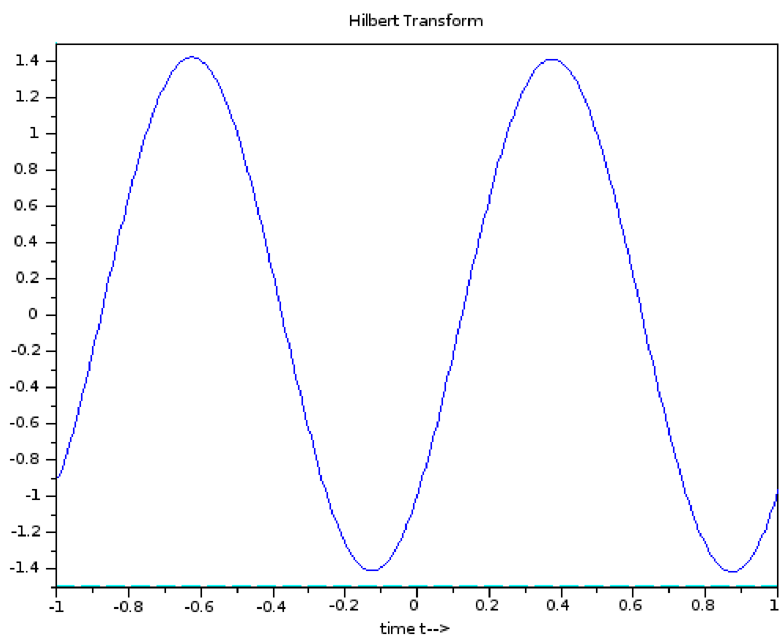

Figure 11.6: Hilbert Transform

```
clear;
n=1;
for t=-1:0.01:1
    xr(n)=exp(%i*2*%pi*t);
    n=n+1;
end
//Computing Hilbertb Transform
x1=hilbert(real(xr));
x2=hilbert(imag(xr));
x=x1+x2;
t=-1:0.01:1;
plot(t,xr);
title('Given Signal x(t)');
xlabel('time t--->');
figure(1);
t=-1:0.01:1;
plot(t,imag(x));
title('Hilbert Transform');
xlabel('time t--->');
```

Scilab code Exa 11.6 Hilbert Transform

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

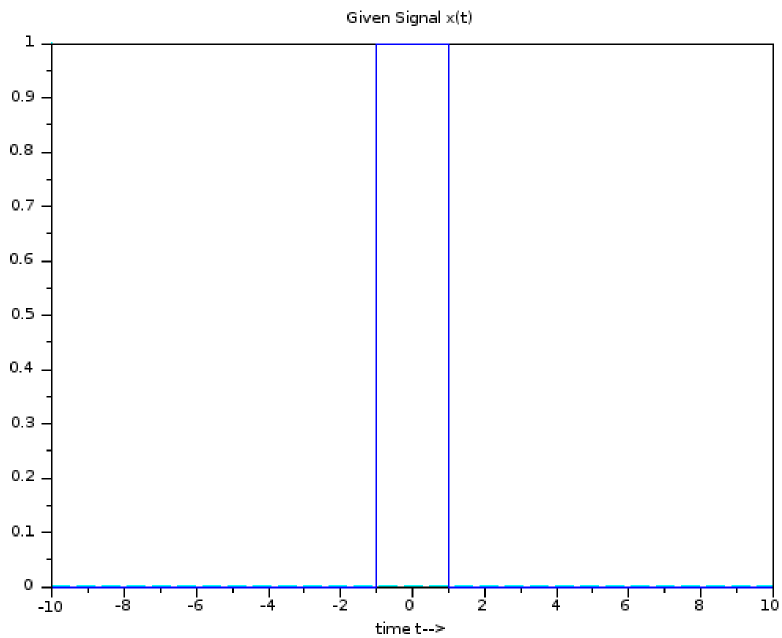

Figure 11.7: Hilbert Transform

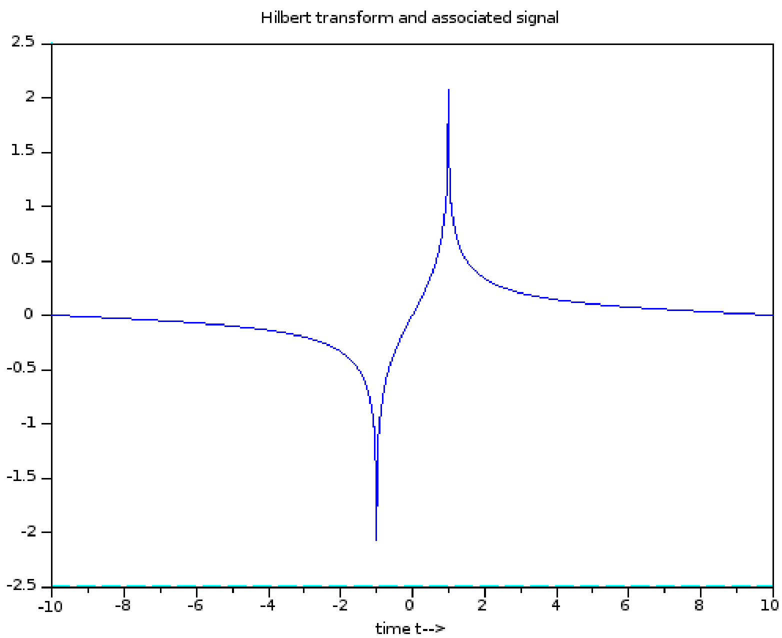

Figure 11.8: Hilbert Transform

```
for t=-10:0.01:10;
    if t>-T/2 & t<T/2 then
        x(n)=A;
        n=n+1;
    else x(n)=0;
        n=n+1;
    end
end
t=-10:0.01:10;
//Signal x(t)
plot(t,x);
title('Given Signal x(t)');
xlabel('time t—>');
//Computing Hilbert Transform
xr=hilbert(x);
xr_imag=imag(xr);
figure(1);
t=-10:0.01:10;
plot(t,xr_imag);
title('Hilbert transform and associated signal');
xlabel('time t—>');
```