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

Created by
Shafquet Iftekhar
B.Tech
Electronics Engineering
Jamia Millia Islamia
College Teacher
Dr. Sajad A. Loan
Cross-Checked by

July 31, 2019

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

Book Description

Title: Signals And Systems

Author: S. Ghosh

Publisher: Pearson Education, New Delhi

Edition: 1

Year: 2007

ISBN: 81-7758-380-8

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

Fundamentals of Signals and Systems

Scilab code Exa 1.1 Time shifting and scaling

```
//Example1.1
clc;
t=0:0.01:9;
A=0:4/900:4;
for i=1:length(t)
    if t(i)<3 then
        x(i)=A(i)*t(i);
    else
        x(i)=0
    end
end
t1=t+3;
subplot(2,2,1)
plot(t1,x);
xtitle('x(t-3)');
subplot(2,2,2)
plot(4*t,x);
```

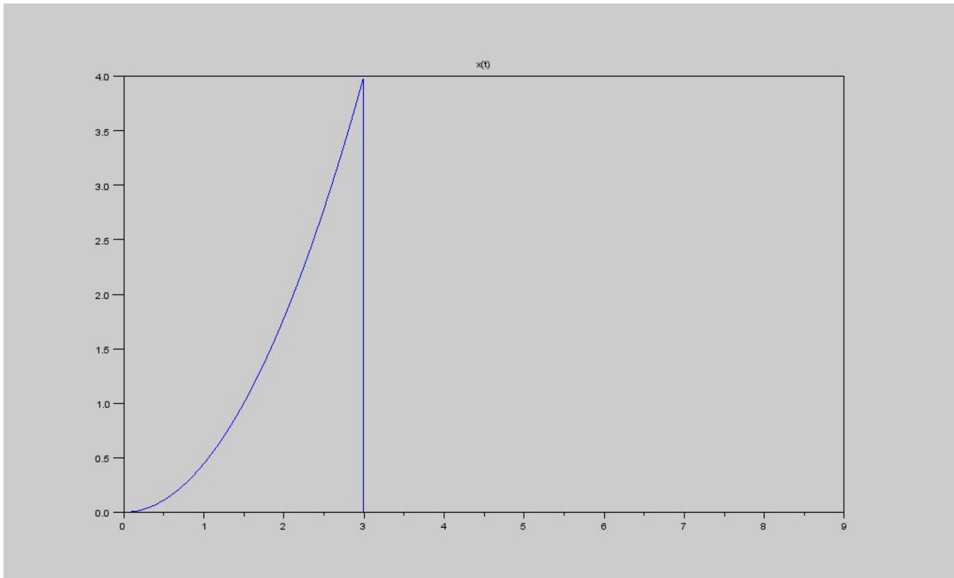

Figure 1.1: Time shifting and scaling

```
xtitle('x(t/4)');
subplot(2,2,3)
plot(t/3,x);
xtitle('x(3t)');
subplot(2,2,4)
t2=-9:0.01:0
plot(t2,x($:-1:1));
xtitle('x(-t)');
figure
plot(t,x);
xtitle('x(t)');
```

Scilab code Exa 1.3.a `Check for energy or power signal`

```
//Example 1.3a
```

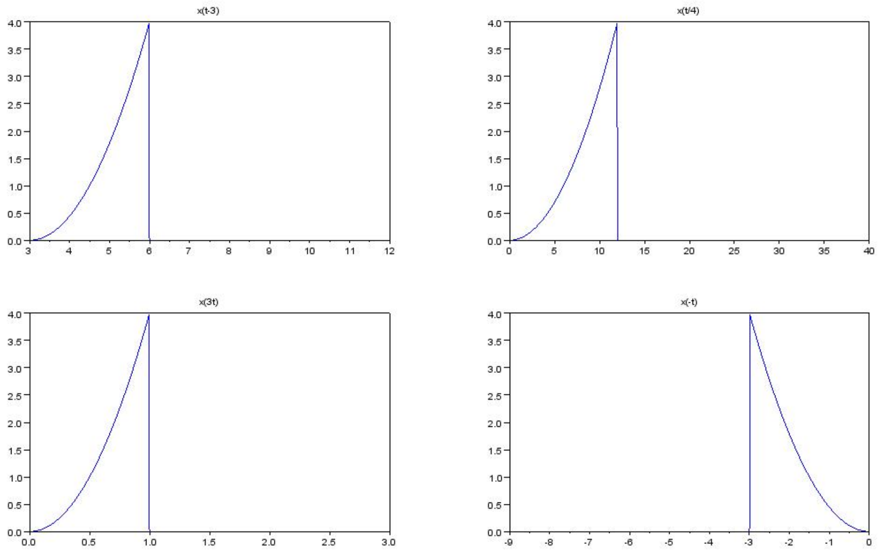

Figure 1.2: Time shifting and scaling

```
//Find whether the the given signal is energy or power signal
clc;
A=0.5;
phi=0;
t=0:0.001:10;
y=A*sin(2*%pi*t+phi);
P=(integrate('A^2*(sin(2*%pi*t))^2','t',0,2*%pi))/(2*%pi);
disp(P,'Power of the signal is');
disp('Since the power of the given signal is finite so we can say that this signal is a power signal');
```

Scilab code Exa 1.3.b `Check for energy or power signal`

```
//Example 1.3b
```

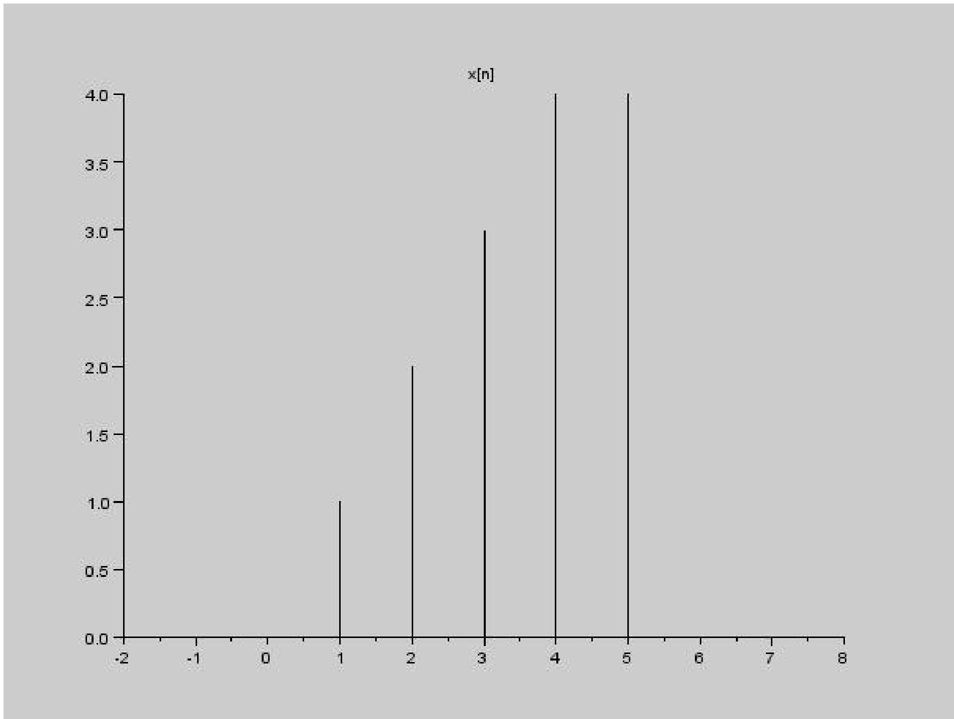

Figure 1.3: Time shifting and scaling

```
//Find whether the the given signal is energy or
   power signal
clc;
b=0.5;
t=0:0.001:10;
y=exp(-b*t);
E=integrate('(exp(−b∗t))^2','t',0,2*%pi);
disp(E,'Energy of the signal is');
disp('Since the energy of the given signal is finite
    so we can say that this signal is an energy
   signal');
```

Scilab code Exa 1.4 Time shifting and scaling

```
//Example 1.4
//Time Shifting And Scaling
clc;
n=-2:8;
x=[0,0,0,1,2,3,4,4,0,0,0];
n1=n+3;
subplot(2,2,1);
plot2d3(n1,x);
xtitle('x[n-3]');
subplot(2,2,2);
plot2d3(ceil(n/3),x);
xtitle('x[3n]');
subplot(2,2,3);
n2=-8:2;
plot2d3(n2,x($:-1:1));
xtitle('x[-n]');
subplot(2,2,4)
n3=n2+3;
plot2d3(n3,x($:-1:1));
xtitle('x[-n+3]');
figure
plot2d3(n,x);
xtitle('x[n]');
```

Scilab code Exa 1.5 Sum and multiplication of two signal

```
//Example 1.5
clc;
n=-1:5;
x1=[0,0,1,2,-3,0,-2];
x2=[2,1,-1,3,2,0,0];
```

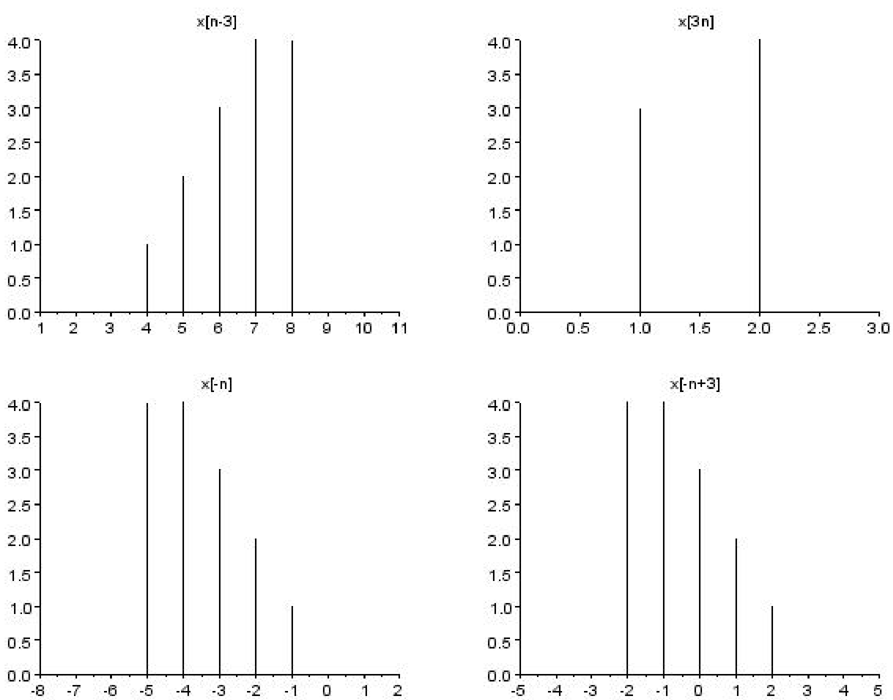

Figure 1.4: Time shifting and scaling

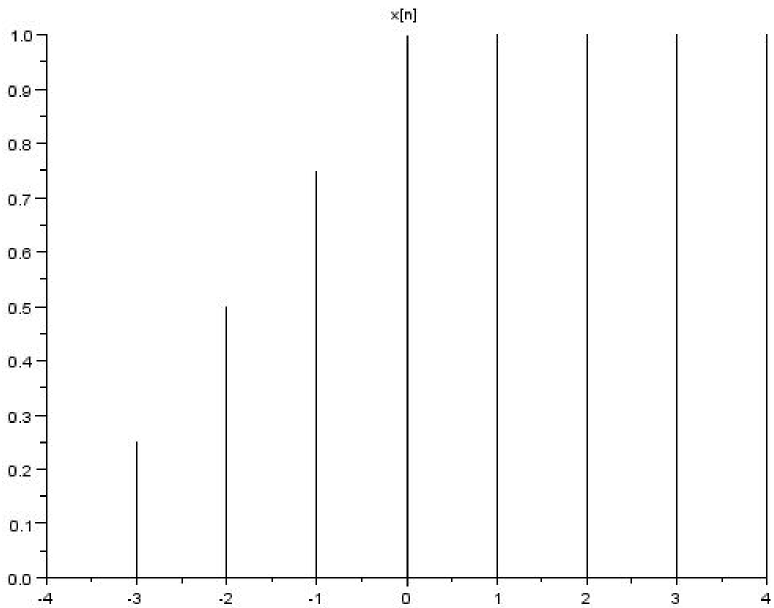

Figure 1.5: Ploting of signal

```
y1=x1+x2;
disp(y1,'y1[n]=');
y2=x1.*x2;
disp(y2,'y2[n]=');
```

Scilab code Exa 1.6 `Ploting of signal`

```
//Example 1.6
clc;
n1=-4:-1;
for i=1:length(n1)
```

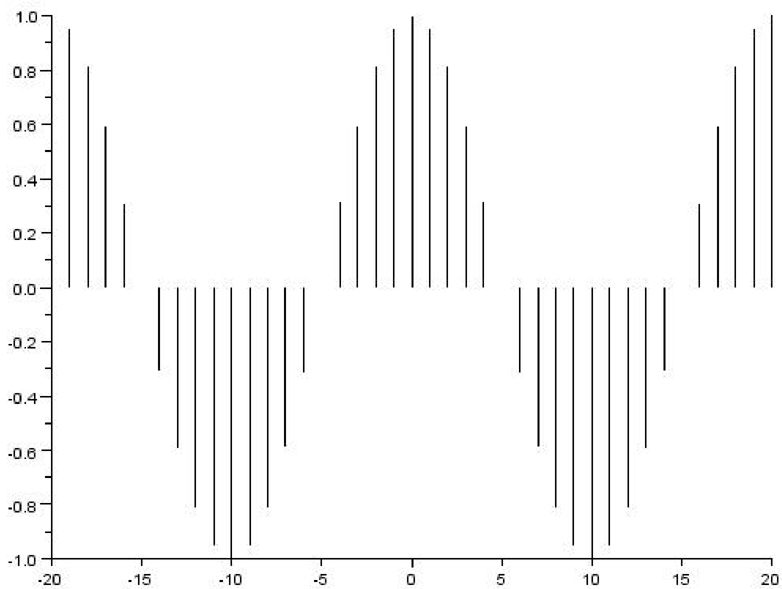

Figure 1.6: Check for periodicity

```
    x(i)=1+(n1(i)/4);
end
for j=5:9
    x(j)=1;
end
n=-4:4;
plot2d3(n,x);
xtitle('x[n]');
```

Scilab code Exa 1.8.a `Check for periodicity`

```
//Example 1.8a
//Check whether the given signal is periodic or not
clc;
n=-20:20;
x=cos(0.1*%pi*n);
plot2d3(n,x);
f=0.1*%pi/(2*%pi);//f is no. of cycles per sample
N=1/f;//N is no. of samples per cycle
disp('samples',N,'Figure shows that the signal is
    periodic with period equal to');
```

Scilab code Exa 1.9.a `Find energy of signal`

```
//Example 1.9a
clc;
E=0;
for n=0:100
    x(n+1)=(-0.3)^n;
end
for n=0:100
    E=E+x(n+1)^2;
end
if E<%inf then
    disp(E,'The given signal is energy signal with
        energy=');
else
    disp('The given signal is not energy signal');
end
```

Scilab code Exa 1.9.b `Find power of signal`

```
//Example 1.9b
clc;
```

```
for n=0:100
    x(n+1)=2;
end
P=0;
for n=0:100
    P=P+(abs(x(n+1)^2))/100;
end
if P<%inf then
    disp(P,'The given signal is power signal with
       power =');
else
    disp('The given signal is not power signal');
end
```

Scilab code Exa 1.9.c `Find power of signal`

```
//Example 1.9c
clc;
for n=1:100
    x(n)=3*exp(%i*2*n);
end
P=0;
for n=1:100
    P=P+(abs(x(n)^2))/100;
end
if P<%inf then
    disp(P,'The given signal is power signal with
       power =');
else
    disp('The given signal is not power signal');
end
```

Scilab code Exa 1.11.b `Check for time invariant systems`

```
//Example 1.11b
//Determine whether the following signal is time
   invariant or not
clc;
n0=2;
N=10;
for n=1:N
    x(n)=n;
    y(n)=n*x(n);
end
inputshift=x(N-n0);
outputshift=y(N-n0);
if(inputshift==outputshift)
    disp('THE GIVEN SYSTEM IS TIME INVARIANT')
else
    disp('THE GIVEN SYSTEM IS TIME VARIANT');
end
```

Scilab code Exa 1.12.b Check for linear systems

```
//Example 1.12b
//Determine whether the system is linear or not
clc;
x1=[1,1,1,1]
x2=[2,2,2,2]
a=1
b=1
for n=1:length(x1)
    x3(n)=a*x1(n)+b*x2(n)
end
for n=1:length(x1)
    y1(n)=x1(n)^2
    y2(n)=x2(n)^2
    y3(n)=x3(n)^2
end
```

```
for n=1:length(y1)
    z(n)=a*y1(n)+b*y2(n)
end
count=0
for n=1:length(y1)
    if(y3(n)==z(n))
        count=count+1;
     end
end
if(count==length(y3))
disp('It satisfy the superposition principle');
disp('THE GIVEN SYSTEM IS LINEAR ');
else
    disp('It does not satisfy superposition
       principle ');
    disp('THE GIVEN SYSTEM IS NON LINEAR');
end
```

Scilab code Exa 1.15.a Check for periodicity

```
//Example 1.15a
//Check whether the given signal is periodic or not
clc;
t=-10:0.01:10;
y=cos(t+(%pi/3));
plot(t,y);
disp('Plot shows that the given signal is periodic
   with period 2*%pi');
```

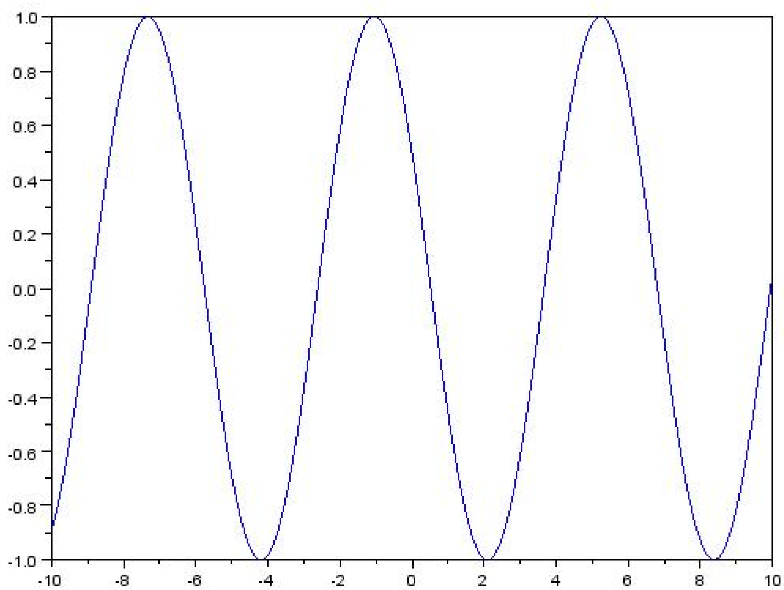

Figure 1.7: Check for periodicity

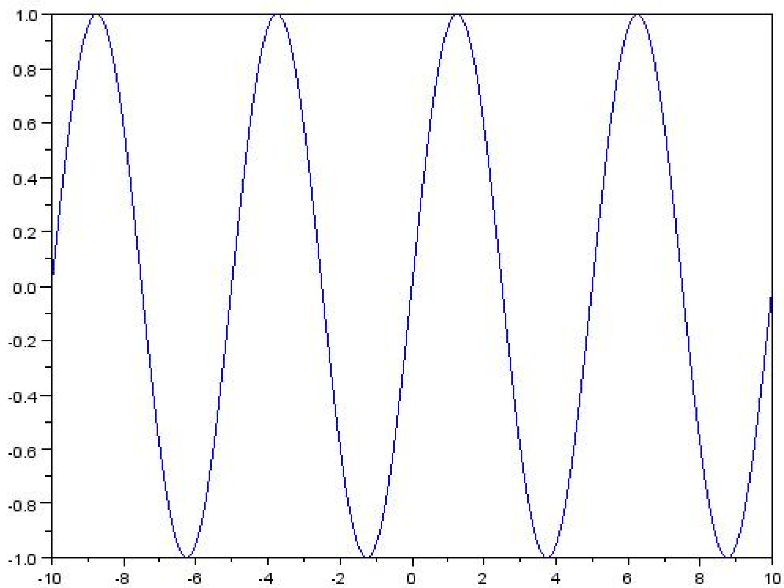

Figure 1.8: Check for periodicity

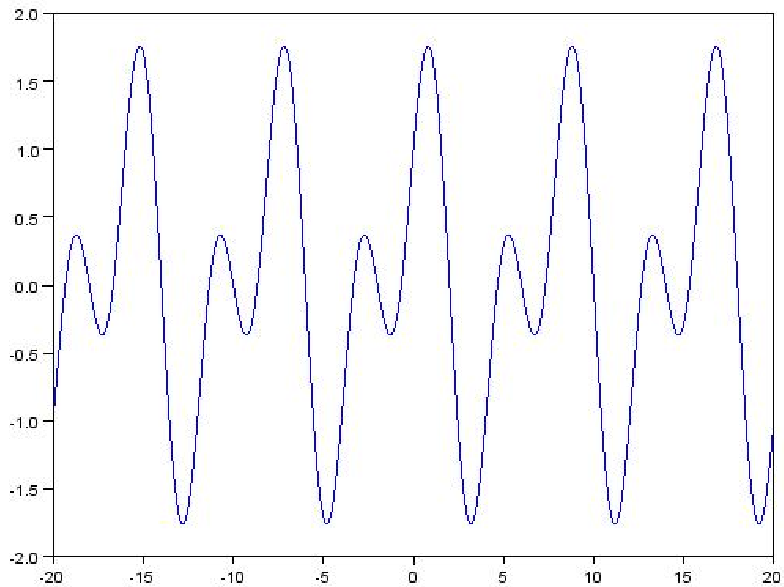

Figure 1.9: Check for periodicity

Scilab code Exa 1.15.b `Check for periodicity`

```
//Example 1.15b
//Check whether the given signal is periodic or not
clc;
t=-10:0.01:10;
y=sin((2*%pi/5)*t);
plot(t,y);
disp('Plot shows that the given signal is periodic
   with period 5');
```

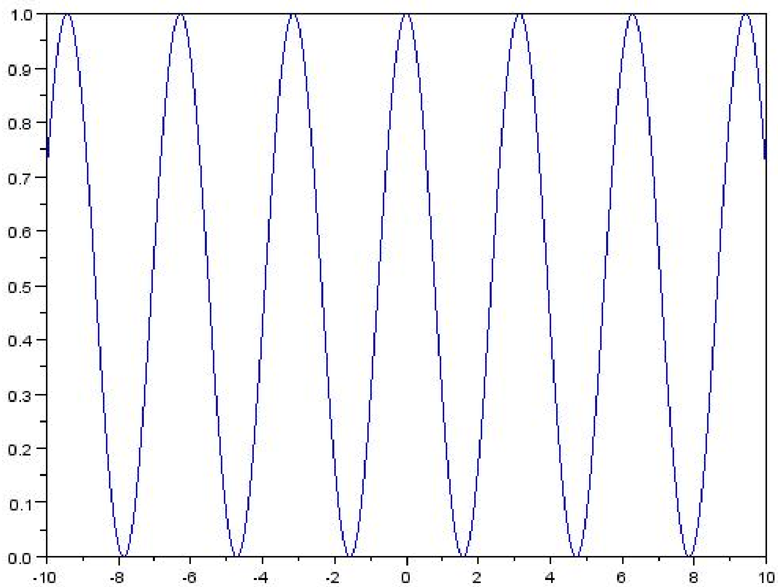

Figure 1.10: Check for periodicity

Scilab code Exa 1.15.c `Check for periodicity`

```
//Example 1.15c
//Check whether the given signal is periodic or not
clc;
t=-20:0.01:20;
y=sin((%pi/2)*t)+cos((%pi/4)*t);
plot(t,y);
disp('Plot shows that the given signal is periodic
   with period 40');
```

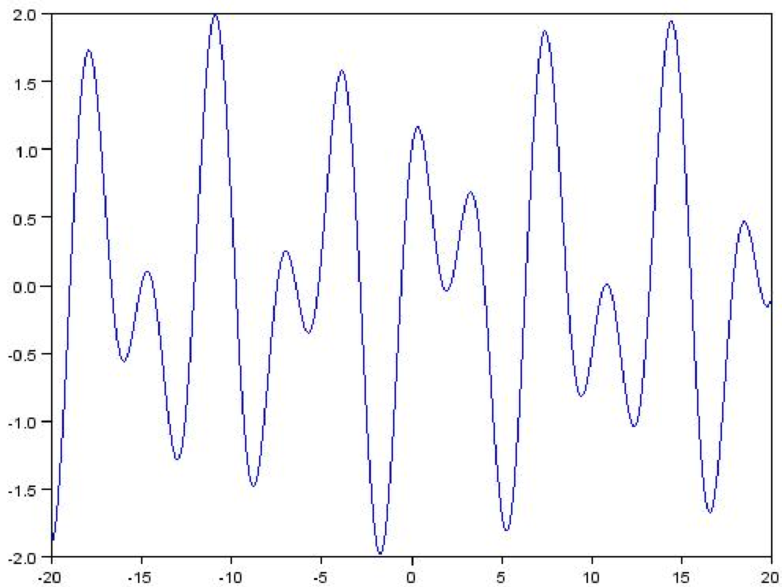

Figure 1.11: Check for periodicity

Scilab code Exa 1.15.d `Check for periodicity`

```
//Example 1.15d
//Check whether the given signal is periodic or not
clc;
t=-10:0.01:10;
y=(cos(t))^2;
plot(t,y);
disp('Plot shows that the given signal is periodic
   with period %pi');
```

Scilab code Exa 1.15.e Check for periodicity

```
//Example 1.15e
//Check whether the given signal is periodic or not
clc;
t=-20:0.01:20;
y=sin(t)+cos(sqrt(3)*t);
plot(t,y);
disp('Plot shows that the given signal is NOT
    periodic');
```

Scilab code Exa 1.19.a Check for periodicity

```
//Example 1.19a
//Check whether the given signal is periodic or not
clc;
n=-50:50;
y=sin(n/5);
plot2d3(n,y);
disp('Plot shows that the given signal is periodic')
    ;
```

Scilab code Exa 1.19.b Check for periodicity

```
//Example 1.19b
//Check whether the given signal is periodic or not
clc;
n=-20:20;
x=exp(%i*(%pi/5)*n);
plot2d3(n,x);
disp('Plot shows that the given signal is periodic')
    ;
```

Scilab code Exa 1.19.c `Check for periodicity`

```
//Example 1.19c
//Check whether the given signal is periodic or not
clc;
n=-75:75;
x=cos((%pi/5)*n)+sin((%pi/6)*n);
plot2d3(n,x);
disp('Plot shows that the given signal is periodic')
    ;
```

Chapter 2

Fourier Series

Scilab code Exa 2.2 Find trigonometric fourier series

```
clear ;
close;
clc;
T0=4;
t=.01:0.01:2*T0;
t_temp=0.01:0.01:T0/2;
s=length(t)/length(t_temp);
x=[];
for i=1:s
    if modulo(i,2)==0 then
        x=[x zeros(1,length(t_temp))];
    else
        x=[x ones(1,length(t_temp))];
    end
end
a=gca();
plot(t,x)
poly1=a.children.children;
poly1.thickness=3;
```

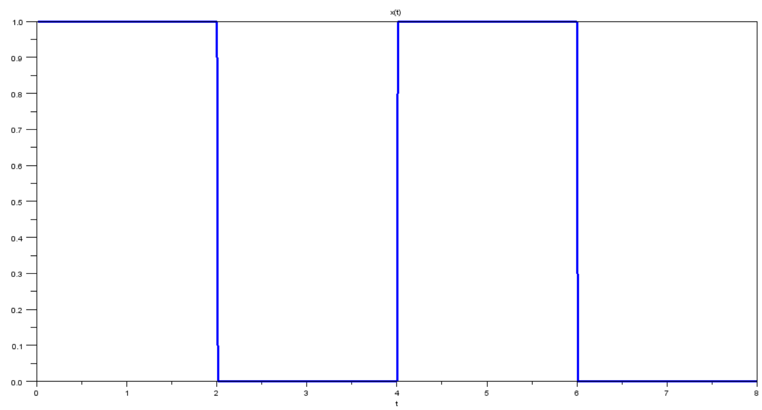

Figure 2.1: Find trigonometric fourier series

```
poly1.foreground=2;
xtitle('x(t)','t')
w0=%pi/2;
for k=1:5
    cc(k,:)=exp(-%i*k*w0*t);
    ck(k)=x*cc(k,:)'/length(t);
    if abs(ck(k))<0.01 then
    ck(k)=0;
    else if imag(ck(k))<0.01 then
    ck(k)=real(ck(k));
    end
    end

end
a=2*real(ck);
b=2*imag(ck);
disp(b,'bn=');
```

Chapter 3

Fourier Transform

Scilab code Exa 3.4 Find system function

```
//Example 3.4
//Find system function and output of the system
clc;
syms t;
h=%e^(-3*t);
H=laplace(h,t,'jw');
disp(H,'SYSTEM FUNCTION=');
x=%e^(-2*t);
X=laplace(x,t,'jw');
Y=H*X;
y=ilaplace(Y,'jw',t);
disp(y,'OUTPUT OF THE SYSTEM FOR THE GIVEN INPUT IS=
    ');
```

Scilab code Exa 3.5 Find impulse responce

```
//Example 3.5
//Find the impulse responce and output of the system
```

```
clc;
syms jw t;
H=(jw+1)/((jw+2)*(jw+3));
h=ilaplace(H,jw,t);
disp(h,'IMPULSE RESPONCE=');
x=%e^(-2*t);
X=laplace(x,t,jw);
Y=H*X;
y=ilaplace(Y,jw,t);
disp(y,'OUTPTU OF THE SYSTEM IS=');
```

Chapter 4

Laplace Transform

Scilab code Exa 4.3 Laplace transform of function

```
//Example 4.3
//Laplace transform of f(t)=3-2%e^(-4t)
clc;
syms t;
f=3-2*%e^(-4*t);
F=laplace(f);
disp(F);
```

Scilab code Exa 4.4 Laplace transform of function

```
//Example 4.4
//Laplace transform of f(t)=5cos(wt)+4sin(wt)
clc;
syms w t;
f=5*cos(w*t)+4*sin(w*t);
F=laplace(f);
disp(F);
```

Scilab code Exa 4.8 Laplace transform of function

```
//Example 4.8
//Laplace transform of x(t)=%e^(3t)u(-t)+%e^(t)u(t)
clc;
syms t;
x1=%e^(3*t);
x2=%e^t;
X1=laplace(x1);
X2=laplace(x2);
X=X2-X1;//since x1 is form -%inf to 0
disp(X);
```

Scilab code Exa 4.11 Inverse laplace transform

```
//Example 4.11
clc;
syms s;
I=(3*s+4)/(s^2+4*s);
i=ilaplace(I);
disp(i);
```

Scilab code Exa 4.14 Inverse laplace transform

```
//Example 4.14
clc;
syms s;
F=(s+3)/(s*(s+1)*(s+2));
f=ilaplace(F);
disp(f);
```

Scilab code Exa 4.15 Inverse laplace transform

```
//Example 4.15
clc;
syms s;
F=(s+3)/(s*((s+1)^2)*(s+2));
f=ilaplace(F);
disp(f);
```

Scilab code Exa 4.16 Inverse laplace transform

```
//Example 4.16
clc;
syms s;
F=1/((s^2)*(s+2));
f=ilaplace(F);
disp(f);
```

Scilab code Exa 4.17 Inverse laplace transform

```
//Example 4.17
clc;
syms s;
I=(s+6)/(s*(s+3));
i=ilaplace(I);
Io=limit(s*I,s,0);
disp(Io,'FINAL VALUE OF i(t)');
```

Scilab code Exa 4.18 Inverse laplace transform

```
//Example 4.18
clc;
syms s;
I=(2*s+3)/((s+1)*(s+3));
i=ilaplace(I);
io=limit(i,t,0);
disp(io,'INITIAL VALUE OF i(t)');
```

Scilab code Exa 4.19 Inverse laplace transform

```
//Example 4.19
clc;
syms s;
F=1/((s+1)*(s+2));
f=ilaplace(F);
disp(f);
```

Scilab code Exa 4.28 Determine the input for given output

```
//Example 4.28
clc;
syms t;
h=%e^(-2*t)+%e^(-3*t);
vo=t*%e^(-2*t);
Vo=laplace(vo);
H=laplace(h);
Vi=Vo/H;
```

```
vi=ilaplace(Vi);
disp(vi);
```

Scilab code Exa 4.29 Find unit step responce of system

```
//Example 4.29
clc;
syms t;
h=0.24*(%e^(-0.36*t)-%e^(-2.4*t));
H=laplace(h);
x=1;
X=laplace(x);
Y=X*H;
y=ilaplace(Y);
disp(y);
```

Chapter 5

System Modelling

Scilab code Exa 5.2 Find transfer function

```
//Example 5.2
clc;
syms t;
h=%e^(-3*t);
H=laplace(h);
disp(H,'Transfer Function is');
```

Scilab code Exa 5.3 Find responce of system

```
//Example 5.3
clc;
syms t;
h=%e^(-3*t);
x=%e^(-4*t);
H=laplace(h);
X=laplace(x);
Y=X*H;
y=ilaplace(Y);
disp(y,'y(t)=');
```

Chapter 6

Z Transform

Scilab code Exa 6.1.a `z transform of sequence`

```
//Example 6.1a
clc;
function[za]=ztransfer(sequence,n)
    z=poly(0,'z','r')
    za=sequence*(1/z)^n'
endfunction
x=[1,2,3,4,5,6,7];
n1=0:length(x)-1;
X=ztransfer(x,n1);
disp(X,'X(z)=');
funcprot(0);
```

Scilab code Exa 6.1.b `z transform of sequence`

```
//Example 6.1b
clc;
function[za]=ztransfer(sequence,n)
    z=poly(0,'z','r')
```

```
    za=sequence*(1/z)^n'
endfunction
x=[1,2,3,4,5,6,7];
n1=-2:length(x)-3;
X=ztransfer(x,n1);
disp(X,'X(z)=');
funcprot(0);
```

Scilab code Exa 6.2 Convolution of two sequences

```
//Example 6.2
clc;
function[za]=ztransfer(sequence,n)
    z=poly(0,'z','r')
    za=sequence*(1/z)^n'
endfunction
x1=[1,-3,2];
n1=0:length(x1)-1;
X1=ztransfer(x1,n1);
x2=[1,2,1];
n2=0:length(x2)-1;
X2=ztransfer(x2,n2);
X=X1*X2;
disp(X,'X(z)=');
z=poly(0,'z');
X=[1;-z^-1;-3*z^-2;z^-3;2*z^-4];
n=0:4;
ZI=z^n';
x=numer(X.*ZI);
disp(x,'x[n]=');
```

Scilab code Exa 6.5 z transform

```
//Example 6.5
clc;
syms z n;
x1=4*(5^n);
x2=3*(4^n);
X1=symsum(x1*(z^-n),n,0,%inf);
X2=symsum(x2*(z^-n),n,0,%inf);
X=X1-X2;
```

Scilab code Exa 6.6.a z transform

```
//Example 6.6a
clc;
syms z n;
x=(1/3)^n;
X=symsum(x*(z^-n),n,-%inf,0);
```

Scilab code Exa 6.6.b z transform

```
//Example 6.6b
clc;
syms z n;
x=(1/3)^n;
X1=symsum(x*(z^-n),n,0,%inf);
X2=symsum(x*(z^-n),n,8,%inf);
X=X1-X2;
```

Scilab code Exa 6.12 z transform of sequence

```
//Example 6.12
```

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

Scilab code Exa 6.14.a `z transform`

```
//Example 6.14a
clc;
syms z n;
x1=(1/4)^n;
x2=(1/5)^n;
X1=symsum(x1*(z^-n),n,0,%inf);
X2=symsum(x2*(z^-n),n,0,%inf);
X=X1+X2;
disp(X,'X(z)=');
```

Scilab code Exa 6.14.b `z transform`

```
//Example 6.14b
clc;
syms z n;
x1=(1/5)^n;
x2=(1/4)^n;
X1=symsum(x1*(z^-n),n,0,%inf);
X2=symsum(x2*(z^-n),n,-%inf,-1);
X=X1+X2;
```

```
disp(X,'X(z)=');
```

Scilab code Exa 6.14.c z transform

```
//Example 6.14c
clc;
syms z n;
x1=(1/4)^n;
x2=(1/5)^n;
X1=symsum(x1*(z^-n),n,0,%inf);
X2=symsum(x2*(z^-n),n,-%inf,-1);
X=X1+X2;
disp(X,'X(z)=');
```

Chapter 7

Convolution

Scilab code Exa 7.2 Convolution of two periodic signals

```
//Example 7.2
//Convolution of two periodic signals
clc;
x1=[1,2,3,4];
x2=[3,1,1,3];
X1=fft(x1,-1);
X2=fft(x2,-1);
X3=X1.*X2;
x3=fft(X3,1);
disp(x3,'Convolution of the two given periodic
    signals is');
```

Scilab code Exa 7.3 Linear and circular convolution

```
//Example 7.3
//Linear and circular convolution of two sequences
clc;
x1=[1,2,3,4];
```

```
x2=[3,1,4,2];
y1=convol(x1,x2);
disp(y1,'Linear convolution of the two sequences')
X1=fft(x1,-1);
X2=fft(x2,-1);
Y2=X1.*X2;
y2=fft(Y2,1);
disp(y2,'Circular convolution of the two sequences')
    ;
```

Scilab code Exa 7.4 Convolution of two sequences

```
//Example 7.4
//Convolution of given sequences
clc;
x=[1,2,3,4];
y=[1,-2];
X=convol(x,y);
disp(X,'Convolution of given sequences');
```

Scilab code Exa 7.5 Convolution of two sequences

```
//Example 7.5
//Convolution of two signals
clc;
x=[1,3,2];
y=[4,1,2];
X=convol(x,y);
disp(X,'Convolution of the given sequences');
```

Scilab code Exa 7.6 Convolution of two sequences

```
//Example 7.6
//Convolution of given sequences
clc;
x=[1,-2,2];
y=[2,5,3,6];
X=convol(x,y);
disp(X,'Convolution of the given sequences');
```

Chapter 8

Stability

Scilab code Exa 8.3 Check the stability

```
//Example 8.3
clc;
// Define the polynomial
s=poly(0,"s");
p=3+10*s+5*s^2+s^3;
// Calculate the routh of above polynomial
r=routh_t(p);
disp(r,"Routh array=");
A=r(:,1);
c=0;
x1=0;
eps=0;
for i=1:4
   x1=A(i,1);
   if x1<0
     c=c+1;
   end
end
   if(c>=1) then
       printf("system is unstable");
   else
```

```
        printf("system is stable");
    end
x=roots(p);
```

Scilab code Exa 8.4 Check the stability

```
//Example 8.4
clc;
// Define the polynomial
s=poly(0,"s");
p=10+3*s+2*s^2+s^3;
// Calculate the routh of above polynomial
r=routh_t(p);
disp(r,"Routh array=");
A=r(:,1);
c=0;
x1=0;
eps=0;
for i=1:4
    x1=A(i,1);
    if x1<0
        c=c+1;
    end
end
   if(c>=1) then
       printf("system is unstable");
   else
       printf("system is stable");
   end
x=roots(p);
```

Scilab code Exa 8.5.a Check the stability

```
//Example 8.5a
clc;
// Define the polynomial
s=poly(0,"s");
p=20+36*s+21*s^2+21*s^3+s^4;
// Calculate the routh of above polynomial
r=routh_t(p);
disp(r,"Routh array=");
A=r(:,1);
c=0;
x1=0;
eps=0;
for i=1:5
   x1=A(i,1);
   if x1<0
       c=c+1;
   end
end
   if(c>=1) then
       printf("system is unstable");
   else
       printf("system is stable");
   end
x=roots(p);
```

Scilab code Exa 8.5.b Check the stability

```
//Example 8.5b
clc;
// Define the polynomial
s=poly(0,"s");
p=1+s+2*s^2+3*s^3+6*s^4+s^5;
// Calculate the routh of above polynomial
r=routh_t(p);
disp(r,"Routh array=");
```

```
A=r(:,1);
c=0;
x1=0;
eps=0;
for i=1:6
     x1=A(i,1);
     if x1<0
         c=c+1;
     end
end
   if(c>=1) then
       printf("system is unstable");
   else
       printf("system is stable");
   end
x=roots(p);
```

Scilab code Exa 8.6 Check the stability

```
//Example 8.6
clc;
// Define the polynomial
s=poly(0,"s");
p=1+3*s+8*s^2+4*s^3+2*s^4+s^5;
// Calculate the routh of above polynomial
r=routh_t(p);
A=r(:,1);
c=0;
x=0;
for i=1:6
    x=A(i,1);
    if x<>0
        c=c+1;
    end
end
```

```
    if(c>=1) then
        printf("system is unstable");
    else
        printf("system is stable");
    end
```

Scilab code Exa 8.7 Check the stability

```
//Example 8.7
clc;
// Define the polynomial
s=poly(0,"s");
p=8+4*s+4*s^2+2*s^3+2*s^4+s^5;
// Calculate the routh of above polynomial
r=routh_t(p);
S=roots(p);
disp(r,"Routh array=");
disp(S,"Roots=");
A=r(:,1);
c=0;
x=0;
for i=1:5
    x=A(i,1);
    if x<0
        c=c+1;
    end
end
if(c>=1) then
   printf("system is unstable");
else
    l=length(S);
    c=0;
    for i=1:l
        a=S(i,1);
        r=real(a);
```

```
        if r<0 then
            c=c+1;
        end
    end
    if c==0 then
        printf("system is stable");
    else
        printf("system is unstable");
    end
end
```

Scilab code Exa 8.9 Check the stability

```
//Example 8.9
clc;
// Define the polynomial
s=poly(0,"s");
p=8+4*s+3*s^2+3*s^3+s^4+s^5;
// Calculate the routh of above polynomial
r=routh_t(p);
A=r(:,1);
c=0;
x=0;
for i=1:6
    x=A(i,1);
    if x<>0
        c=c+1;
    end
end
if(c>=1) then
    printf("system is unstable");
else
    printf("system is stable");
end
```

Chapter 11

Discrete Fourier Transform and Fast Fourier Transform

Scilab code Exa 11.1 `DFT of sequence`

```
//Example 11.1
//Find the DFT of x[n]=[1,2,3,4]
clc;
x=[1,2,3,4];
X=fft(x,-1);
disp(X,'X(k)=');
```

Scilab code Exa 11.2 `Circular convolution`

```
//Example 11.2
//Find the circular convolution
clc;
x1=[3,1,3,1];
x2=[1,2,3,4];
X1=fft(x1,-1);
X2=fft(x2,-1);
```

```
X3=X1.*X2;
x3=fft(X3,1);
disp(x3,'x3(n)=x1(n)(N)x2(n)');
```