Scilab Textbook Companion for Digital Signal Processing by S. Salivahanan, A. Vallavaraj And C. Gnanapriya[1]

Created by
Priya Sahani
B TECH EXTC
Electrical Engineering
V.J.T.I
College Teacher
Rizwn Ahmed
Cross-Checked by
Lavitha Pareira

July 17, 2017

[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: Digital Signal Processing

Author: S. Salivahanan, A. Vallavaraj And C. Gnanapriya

Publisher: Tata McGraw - Hill, New Delhi

Edition: 1

Year: 2008

ISBN: 978-0-07-463996-2

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

Chapter 1

Classifications of signals and systems

Scilab code Exa 1.2.a Rectangular wave

```
//Example 1.2 (a)
clc;clear;
t=-5:0.01:5;
x=1*(abs(2*t+3)<0.5);
plot(t,x);
title('x(t)=rect(2t+3)');
```

Scilab code Exa 1.2.b Rectangular wave

```
//Example 1.2 (b)
clc;clear;
t=-5:0.01:5;
x=2*(abs(t-1/4)<0.5);
plot(t,x);
title('x(t)=2*rect(t-1/4)');
```

Scilab code Exa 1.2.c Cosine wave

```
//Example 1.2 (c)
clc;clear;
pi=22/7;
t=-5:0.01:5;
x=cos(2*pi*t-50*pi);
plot(t,x);
title('x(t)=cos(2*pi*t-*pi)');
```

Scilab code Exa 1.2.d Ramp wave

```
//Example 1.2 (d)

clc;clear;
t=-5:0.01:5;
x=-0.5*(t-4);
plot(t,x);
title('x(t)=r(-0.5t+2)');
zoom_rect([-5 0 5 5]);
```

Chapter 2

Fourier Analysis of Preiodic and Aperiodic Continuous Time Signals and Systems

Scilab code Exa 2.1 Fourier Series of Periodic Square Wave

```
//Example 2.1

clc;clear;close;
A=1;T=2;
w0=2*%pi/T;

//Calculation of trignometric fourier series co-efficients
a0=A/T*(integrate('-1','t',-T/2,-T/4)+integrate('+1','t',-T/4,T/4)+integrate('-1','t',T/4,T/2));
for n=1:10;
a(1,n)=2*A/T*(integrate('-cos(n*w0*t)','t',-T/2,-T/4)+integrate('+cos(n*w0*t)','t',-T/4,T/4)+integrate('-cos(n*w0*t)','t',T/4,T/2));
b(1,n)=2*A/T*(integrate('-sin(n*w0*t)','t',-T/2,-T/4)+integrate('+sin(n*w0*t)','t',-T/4,T/4)+integrate('-sin(n*w0*t)','t',T/4,T/2));
```

```
end

//Displaying fourier coefficients
disp(T,'fundamental period T= ',A,'Assumption: Amplitude A= ');
disp('Tignometric fourier series co-efficients:');
disp(a0,'a0= ');disp(a,'an= ');disp(b,'bn= ');

x=[-A*ones(1,25) A*ones(1,50) -A*ones(1,25)]     //Function for ploting purpose
t=-T/2:0.01*T:T/2-0.01;
subplot(311);plot(t,x);
title('x(t)');xlabel('time t');
subplot(312);plot2d3(a);
title('Coefficients an');xlabel('n');
subplot(313);plot2d3(b);
title('Coefficients bn');xlabel('n');
```

Scilab code Exa 2.2 Fourier Series of Periodic Rectangular Wave

```
//Example 2.2

clc;clear;close;
A=1;T=2;
w0=2*%pi/T;

//Calculation of trignometric fourier series co-efficients
a0=A/T*integrate('1','t',-T/4,T/4);
for n=1:10;
a(1,n)=2*A/T*integrate('cos(n*w0*t)','t',-T/4,T/4);
b(1,n)=2*A/T*integrate('sin(n*w0*t)','t',-T/4,T/4);
end

//Displaying fourier coefficients
```

```
disp(T,'fundamental period T= ',A,'Assumption: Amplitude A= ');
disp('Tignometric fourier series co-efficients:');
disp(a0,'a0= ');disp(a,'an= ');disp(b,'bn= ');

x=[zeros(1,25) A*ones(1,50) zeros(1,25)];
t=-T/2:0.01*T:T/2-0.01;
subplot(311);plot(t,x);
title('x(t)');xlabel('time t');
subplot(312);plot2d3(a);
title('Coefficients an');xlabel('n');
subplot(313);plot2d3(b);
title('Coefficients bn');xlabel('n');
```

Scilab code Exa 2.3 Fourier Series of Periodic Half Wave Rectified Sine Wave

```
//Example 2.3

clc;clear;close;
A=1;T=2;
w0=2*%pi/T;

//Calculation of trignometric fourier series co-efficients
a0=A/T*integrate('sin(w0*t)','t',0,T/2);
for n=1:10;
a(1,n)=2*A/T*integrate('sin(w0*t)*cos(n*w0*t)','t',0,T/2);
b(1,n)=2*A/T*integrate('sin(w0*t)*sin(n*w0*t)','t',0,T/2);
end

//Displaying fourier coefficients
disp(T,'fundamental period T= ',A,'Assumption: Amplitude A= ');
```

```
disp('Tignometric fourier series co-efficients:');
disp(a0,'a0= ');disp(a,'an= ');disp(b,'bn= ');

t=0:0.01*T:T/2;
x=[A*sin(w0*t) zeros(1,50)];
t=0:0.01*T:T;
subplot(311);plot(t,x);
title('x(t)');xlabel('time t');
subplot(312);plot2d3(a);
title('Coefficients an');xlabel('n');
subplot(313);plot2d3(b);
title('Coefficients bn');xlabel('n');
```

Scilab code Exa 2.4 Fourier Series of Periodic Triangular Wave

```
//Example 2.4

clc;clear;close;
A=1;T=2;
w0=2*%pi/T;

//Calculation of trignometric fourier series co-efficients
a0=4*A/T*(integrate('t-0.5*T','t',-T/2,-T/4)+integrate('t','t',-T/4,T/4)+integrate('-t+0.5*T','t',T/4,T/2));
for n=1:10;
a(1,n)=2*4*A/T*(integrate('(t-0.5*T)*cos(n*w0*t)','t',-T/2,-T/4)+integrate('t*cos(n*w0*t)','t',-T/4,T/4)+integrate('(-t+0.5*T)*cos(n*w0*t)','t',T/4,T/2));
b(1,n)=2*4*A/T*(integrate('(t-0.5*T)*sin(n*w0*t)','t',-T/2,-T/4)+integrate('t*sin(n*w0*t)','t',-T/4,T/4)+integrate('(-t+0.5*T)*sin(n*w0*t)','t',T/4,T/2));
```

```
end

//Displaying fourier coefficients
disp(T,'fundamental period T= ',A,'Assumption:
    Amplitude A= ');
disp('Tignometric fourier series co-efficients:');
disp(a0,'a0= ');disp(a,'an= ');disp(b,'bn= ');

t=-T/2:0.01*T:T/2;
x=[-4*A/T*t(1:25)-2*A 4*A/T*t(26:75) -4*A/T*t
    (76:101)+2*A];
subplot(311);plot(t,x);
title('x(t)');xlabel('time t');
subplot(312);plot2d3(a);
title('Coefficients an');xlabel('n');
subplot(313);plot2d3(b);
title('Coefficients bn');xlabel('n');
```

Scilab code Exa 2.5 Fourier Series of Periodic Rectangular Pulse

```
//Example 2.5

clc;clear;close;
A=1;T=2;d=0.1;
w0=2*%pi/T;

//Calculation of trignometric fourier series co-
    efficients
a0=A/T*integrate('1','t',-T/4,T/4);
for n=1:10;
a(1,n)=2*A/T*integrate('cos(n*w0*t)','t',-d/2,d/2);
b(1,n)=2*A/T*integrate('sin(n*w0*t)','t',-d/2,d/2);
end

//Displaying fourier coefficients
```

```
disp(d,'pulse width d= ',T,'fundamental period T= ',
    A,'Assumption: Amplitude A= ');
disp('Tignometric fourier series co-efficients:');
disp(a0,'a0= ');disp(a,'an= ');disp(b,'bn= ');

n=round(50*d/T);                                    //
    Variable used for plotting pulses accurately
x=[zeros(1,50-n) A*ones(1,2*n+1) zeros(1,50-n)]
t=-T/2:0.01*T:T/2;
subplot(311);plot(t,x);
title('x(t)');xlabel('time t');
subplot(312);plot2d3(a);
title('Coefficients an');xlabel('n');
subplot(313);plot2d3(b);
title('Coefficients bn');xlabel('n');
```

Scilab code Exa 2.6 Fourier Series of Square Wave

```
//Example 2.6

clc;clear;close;
A=1;T=2;
w0=2*%pi/T;

//Calculation of trignometric fourier series co-
    efficients
a0=A/T*(integrate('-1','t',-T/2,0)+integrate('+1','t
    ',0,T/2));
for n=1:10
a(1,n)=2*A/T*(integrate('-cos(n*w0*t)','t',-T/2,0)+
    integrate('+cos(n*w0*t)','t',0,T/2));
b(1,n)=2*A/T*(integrate('-sin(n*w0*t)','t',-T/2,0)+
    integrate('+sin(n*w0*t)','t',0,T/2));
end
a=clean(a);b=clean(b);               //Function used to
```

```
round small entities to zero

//Calculation of exponential fourier series co-efficients
function y=f(t),y=complex(cos(n*w0*t),-sin(n*w0*t)),endfunction;
for n=-10:10
c(1,n+11)=A/T*(-1*intc(-T/2,0,f)+intc(0,T/2,f));
end
c=clean(c);                              //Function used to round small entities to zero

//Calculation of trignometric fourier series co-efficients from exponential fourie series coefficients
a01=c(1);
a1=2*real(c(12:21));
b1=-2*imag(c(12:21));

//Displaying fourier coefficients
disp(T,'fundamental period T= ',A,'Assumption: Amplitude A= ');
disp('Tignometric fourier series co-efficients:');
disp(a0,'a0= ');disp(a,'an= ');disp(b,'bn= ');
disp('Exponential fourier series co-efficients');
disp(c(11),'c0= ');disp(c(12:21),'cn= ');disp(c(10:-1:1),'c-n= ');
disp('Trignometric fourier series co-efficients from exponential coefficients:');
disp(a01,'a0= ');disp(a1,'an= ');disp(b1,'bn= ');
disp('The co-effifcients obtained are same by both methods')

x=[-A*ones(1,50) A*ones(1,51)];
t=-T/2:0.01*T:T/2;
n=-10:10;
subplot(311);plot(t,x);
title('x(t)');xlabel('time t');
```

```
subplot(312);plot2d3(a);
title('Coefficients an');xlabel('n');
subplot(313);plot2d3(b);
title('Coefficients bn');xlabel('n');
figure;
subplot(311);plot(t,x);
title('x(t)');xlabel('time t');
subplot(312);plot2d3(n,abs(c));
title('Magnitude of Coefficients |c|');xlabel('n');
subplot(313);plot2d3(n,atan(c));
title('Phase of Coefficients /_c');xlabel('n');
```

Scilab code Exa 2.8 Complex fourier series representation

```
//Example 2.8

clc;clear;close;
t=poly(0,'t');
//cn=3/(4+(n*%pi)^2)
Pt=0.669;          //Total energy
Preq=0.999*Pt;     //Required energy
c0=3/(4+(0*%pi)^2);
disp(c0,'c0=');
P=(abs(c0))^2;
c=[];n=0;
while P<Preq
   n=n+1;
   c(n)=3/(4+(n*%pi)^2);
   disp(c(n),'cn=',n,'n=');
   P=P+2*(abs(c(n)))^2;
end
disp(Pt,'Total power Pt=');
disp(Preq,'99.9% of total power Preqd=');
disp(n,'To iclude 99.9% of energy, we need to retain
    n terms where n=');
```

Chapter 3

Applications of Laplace Transform to System Analysis

Scilab code Exa 3.10 Poles and Zeros

```
//Example 3.10

clc;clear;close;
s=poly(0,'s');
I=3*s/(s+2)/(s+4);
disp(I,'Given Transfer Function:');
zero=roots(numer(I));
pole=roots(denom(I));
disp(zero,'Zeros of transfer function: ');
disp(pole,'Poles of transfer function: ');
plzr(I);
```

Scilab code Exa 3.11 Poles and zeros

```
//Example 3.11

```

```
clc;clear;close;
s=poly(0,'s');
F=4*(s+1)*(s+3)/(s+2)/(s+4);
disp(F,'Given Transfer Function:');
zero=roots(numer(F));
pole=roots(denom(F));
disp(zero,'Zeros of transfer function: ');
disp(pole,'Poles of transfer function: ');
plzr(F);
```

Scilab code Exa 3.12 Poles and zeros

```
//Example 3.12

clc;clear;close;
s=poly(0,'s');
F=10*s/(s^2+2*s+2);
disp(F,'Given Transfer Function:');
zero=roots(numer(F));
pole=roots(denom(F));
disp(zero,'Zeros of transfer function: ');
disp(pole,'Poles of transfer function: ');
plzr(F);
```

Chapter 4

Z Transforms

Scilab code Exa 4.2 Z transform

```
//Example 4.2

clc;clear;close;
z=poly(0,'z');
x1=[3 1 2 5 7 0 1];n1=-3:3;
X1=x1*(z^-n1)';
x2=[2 4 5 7 0 1 2];n2=-2:4;
X2=x2*(z^-n2)';
x3=[1 2 5 4 0 1]; n3=0:5;
X3=x3*(z^-n3)';
x4=[0 0 1 2 5 4 0 1];n4=0:7;
X4=x4*(z^-n4)';
X5=z^0;
X6=z^-5;
X7=z^5;
disp(X1,'x1(n)={3,1,2,5,7,0,1} X1(z)=');
disp(X2,'x2(n)={2,4,5,7,0,1,2} X2(z)=');
disp(X3,'x3(n)={1,2,5,4,0,1} X3(z)=');
disp(X4,'x4(n)={0,0,1,2,5,4,0,1} X4(z)=');
disp(X5,'x5(n)=delta(n) X5(z)=');
disp(X6,'x6(n)=delta(n-5) X6(z)=');
```

```
disp(X7,'x7(n)=delta(n+5) X7(z)=');
```

Scilab code Exa 4.4 Z transform

```
//Example 4.4

clc;clear;close;
z=poly(0,'z');
x=[1 3 0 0 6 -1];n=-1:4;
X=x*(z^-n)';
disp(X,'x(n)={1,3,0,0,6,-1} X(z)=');
```

Scilab code Exa 4.13 Convolution

```
//Example 4.13

clc;clear;close;
z=poly(0,'z');
x1=[4 -2 1];n1=0:length(x1)-1;
X1=x1*(z^-n1)';
x2=[1 1 1 1 1];n2=0:length(x2)-1;
X2=x2*(z^-n2)';
X3=X1*X2;
l=coeff(numer(X3));
x3=l(:,$:-1:1);
disp(X1,'x1(n)={4,-2,1} X1(z)=');
disp(X2,'x2(n)={1,1,1,1,1} X2(z)=');
disp(X3,'Z transform of convolution of the two
    signals X3(z)=');
disp(x3,'Convolution result of the two signals= ')
```

Scilab code Exa 4.14 Convolution

```
//Example 4.14

clc;clear;close;
z=poly(0,'z');
x1=[2 1 0 0.5];n1=0:length(x1)-1;
X1=x1*(z^-n1)';
x2=[2 2 1 1];n2=0:length(x2)-1;
X2=x2*(z^-n2)';
X3=X1*X2;
l=coeff(numer(X3));
x3=l(:,$:-1:1);
disp(X1,'x1(n)={2,1,0,0.5} X1(z)=');
disp(X2,'x2(n)={2,2,1,1} X2(z)=');
disp(X3,'Z transform of convolution of the two
    signals X3(z)=');
disp(x3,'Convolution result of the two signals= ')
```

Scilab code Exa 4.16 Cross correlation

```
//Example 4.16

clc;clear;close;
z=poly(0,'z');
x1=[1 2 3 4];n1=0:length(x1)-1;
X1=x1*(z^-n1)';
x2=[4 3 2 1];n2=0:length(x2)-1;
X2=x2*(z^-n2)';
X2_=x2*(z^n2)';
X3=X1*X2_;
l=coeff(numer(X3));
x3=l(:,$:-1:1);
disp(X1,'x1(n)={4,-2,1} X1(z)=');
disp(X2,'x2(n)={4,-2,1} X2(z)=');
```

```
disp(X3,'Z transform of cross crrelation of the two signals X3(z)=');
disp(x3,'Cross correlation result of the two signals = ')
```

Scilab code Exa 4.19 System response

```
//Example 4.19

clc;clear;close;
z=poly(0,'z');
h=[1 2 3];n1=0:length(h)-1;
H=h*(z^-n1)';
y=[1 1 2 -1 3];n2=0:length(y)-1;
Y=y*(z^-n2)';
X=Y/H;
l=coeff(numer(X));
x=l(:,$:-1:1);
disp(H,'h(n)={1,2,3} H(z)=');
disp(Y,'y(n)={1,1,2,-1,3} Y(z)=');
disp(X,'Z transform of input sequence X(z)=');
disp(x,'Inpput Sequence = ')
```

Chapter 5

Linear Time Invariant Systems

Scilab code Exa 5.20 System response

```
//Example 5.20

clc;clear;close;
z=poly(0,'z');
x=[-1 1 0 -1];n=0:length(x)-1;
X=x*(z^-n)';
H=0.2-0.5*z^-2+0.4*z^-3
Y=H*X;
l=coeff(numer(Y));
y=l(:,$:-1:1);
disp(X,'Input sequence x(n)={-1,1,0,-1} X(z)=');
disp(H,'System Transfer Function H(z)=');
disp(Y,'Z transform of output response Y(z)=');
disp(y,'Digital output sequence y=')
```

Scilab code Exa 5.21 Poles and zeros

```
//Example 5.21
```

```
clc;clear;close;
z=poly(0,'z');
H=(1+z^-1)/(1+3/4*z^-1+1/8*z^-2);
pole=roots(numer(H));
zero=roots(denom(H));
disp(H,'System Transfer Function H(z)=');
disp(zero,'System zeros are at');
disp(pole,'System poles are at ');
plzr(H);
```

Chapter 6

Discrete and Fast Fourier Transforms

Scilab code Exa 6.1 Linear and Circular convolution

```
//Example 6.1
clc;clear;close;
x1=[1 1 2 2];
x2=[1 2 3 4];
ylength=length(x1);
//Calculation of linear convolution
z=convol(x1,x2);
//Calculation of circular convolution
for n=1:ylength
    y(n)=0;
    for k=1:ylength,
        l=n-k+1;
        if l <= 0 then
            l=l+ylength;
        end
     y(n)=y(n)+(x1(k)*x2(l));
     end
end
//Calculation of circular convolution using DFT and
```

```
   IDFT
X1=fft(x1,-1);
X2=fft(x2,-1);
Y1=X1.*X2;
y1=fft(Y1,1);
y1=clean(y1);
disp(z,'Linear Convolution sequence is z(n): ');
disp(y,'Circular Convolution sequence is y(n): ');
disp(y1,'Circular Convolution sequence calculated
   using DFT-IDFT method is y(n): ');
```

Scilab code Exa 6.2 `FIR filter response`

```
//Example 6.2

clc;clear;close;
x=[1 2];
h=[1 2 4];

//Calculation of linear convolution
y=convol(x,h);
disp(x,'Input Sequence is x(n): ');
disp(h,'Impulse respnose of FIR filter h(n): ');
disp(y,'Output sequence is y(n): ');
subplot(3,1,1);
plot2d3(x);
title('Input Seqence x[n]:');ylabel('Amplitude-->');
   xlabel('n-->')
subplot(3,1,2);
plot2d3(h);
title('Impulse Response h[n]:');ylabel('Amplitude-->
   ');xlabel('n-->')
subplot(3,1,3);
plot2d3(y);
title('Output Seqence y[n]=x[n]*h[n] :');ylabel('
```

```
Amplitude—>');xlabel('n—>')
```

Scilab code Exa 6.3 Convolution

```
//Example 6.3

clc;clear;close;
x=[1 1 1];
h=[1 1 1];

//Calculation of linear convolution
y=convol(x,h);
disp(x,'First Sequence is x(n): ');
disp(h,'Second Sequence is h(n): ');
disp(y,'Output sequence is y(n): ');
subplot(3,1,1);
plot2d3(x);
title('First Seqence x[n]:');ylabel('Amplitude—>');
   xlabel('n—>')
subplot(3,1,2);
plot2d3(h);
title('Second Seqence h[n]:');ylabel('Amplitude—>')
   ;xlabel('n—>')
subplot(3,1,3);
plot2d3(y);
title('Convolution Seqence y[n]=x[n]*h[n] :');ylabel
   ('Amplitude—>');xlabel('n—>')
```

Scilab code Exa 6.4 Convolution

```
//Example 6.4

clc;clear;close;
```

```
a=0.5;
n=1:50;
x=ones(1,50);
h=a^n;

//Calculation of linear convolution
for i=1:50
    y(1,i)=sum(h(1:i));
end
disp('First Sequence is x(n)=u(n) ');
disp(a,'Second Sequence is h(n)=a^n*u(n) where a= ');
disp(y,'Output sequence is y(n): ');
subplot(3,1,1);
plot2d3(x);
title('First Seqence x[n]:');ylabel('Amplitude-->');xlabel('n-->')
subplot(3,1,2);
plot2d3(h);
title('Second Seqence h[n]:');ylabel('Amplitude-->');xlabel('n-->')
subplot(3,1,3);
plot2d3(y);
title('Convolution Seqence y[n]=x[n]*h[n] :');ylabel('Amplitude-->');xlabel('n-->')
```

Scilab code Exa 6.5 Convolution

```
//Example 6.5
clc;clear;close;
x=[1 2 3];xmin=0;nx=xmin:length(x)+xmin-1;
h=[1 2 -2 -1];hmin=-1;nh=length(h)+hmin-1;

//Calculation of linear convolution
y=convol(x,h);
```

```
ymin=xmin+hmin;ny=ymin:length(y)+ymin-1;

disp(x,'First Sequence is x(n): ');
disp(h,'Second Sequence is h(n): ');
disp(y,'Output sequence is y(n): ');
subplot(3,1,1);
plot2d3(nx,x);
title('First Seqence x[n]:');ylabel('Amplitude-->');
    xlabel('n-->')
subplot(3,1,2);
plot2d3(nh,h);
title('Second Seqence h[n]:');ylabel('Amplitude-->')
    ;xlabel('n-->')
subplot(3,1,3);
plot2d3(ny,y);
title('Convolution Seqence y[n]=x[n]*h[n] :');ylabel
    ('Amplitude-->');xlabel('n-->')
```

Scilab code Exa 6.6 Convolution

```
//Example 6.6

clc;clear;close;
x=[1 1 0 1 1];xmin=-2;nx=xmin:length(x)+xmin-1;
h=[1 -2 -3 4];hmin=-3;nh=length(h)+hmin-1;

//Calculation of linear convolution
y=convol(x,h);
ymin=xmin+hmin;ny=ymin:length(y)+ymin-1;

disp(x,'First Sequence is x(n): ');
disp(h,'Second Sequence is h(n): ');
disp(y,'Output sequence is y(n): ');
subplot(3,1,1);
plot2d3(nx,x);
```

```
title('First Seqence x[n]:');ylabel('Amplitude-->');
    xlabel('n-->')
subplot(3,1,2);
plot2d3(nh,h);
title('Second Seqence h[n]:');ylabel('Amplitude-->')
    ;xlabel('n-->')
subplot(3,1,3);
plot2d3(ny,y);
title('Convolution Seqence y[n]=x[n]*h[n] :');ylabel
    ('Amplitude-->');xlabel('n-->')
```

Scilab code Exa 6.8 DFT

```
//Example 6.8

clc;clear;close;
L=3;A=1/4;
x=A*ones(1,L);
//Calculation of DFT
X=fft(x,-1);
X=clean(X);
disp(x,'Given Sequence is x(n): ');
disp(X,'DFT of the Sequence is X(k): ');
```

Scilab code Exa 6.9 DFT

```
//Example 6.9

clc;clear;close;
L=3;A=1/5;
n=-1:1;
x=A*ones(1,L);
//Calculation of DFT
```

```
X=fft(x,-1);
X=clean(X);
disp(x,'Given Sequence is x(n): ');
disp(X,'DFT of the Sequence is X(k): ');
```

Scilab code Exa 6.10 `DFT`

```
//Example 6.10

clc;clear;close;
x=[1 1 2 2 3 3];
//Calculation of DFT
X=fft(x,-1);
X=clean(X);
disp(x,'Given Sequence is x(n): ');
disp(X,'DFT of the Sequence is X(k): ');
subplot(3,1,1);
plot2d3(x);
title('Given Seqence x[n]:');ylabel('Amplitude—&gt;');xlabel('n—&gt;');
subplot(3,1,2);
plot2d3(abs(X));
title('Magnitude Spectrum |X(k)|');xlabel('k—&gt;');
subplot(3,1,3);
plot2d3(atan(X));
title('Phase Spectrum /_X(k)');xlabel('k—&gt;');
```

Scilab code Exa 6.11 `DFT`

```
//Example 6.11
```

```

clc;clear;close;
N=8;A=1/4;
n=0:N-1;
x=A^n;
//Calculation of DFT
X=fft(x,-1);
X=clean(X);
disp(x,'Given Sequence is x(n): ');
disp(N,'N=')
disp(X,'N-point DFT of the Sequence is X(k): ');
```

Scilab code Exa 6.12 DFT

```
//Example 6.12

clc;clear;close;
N=4;
n=0:N-1;
x=cos(%pi/4*n);
//Calculation of DFT
X=fft(x,-1);
X=clean(X);
disp(x,'Given Sequence is x(n): ');
disp(X,'DFT of the Sequence is X(k): ');
```

Scilab code Exa 6.13 Inverse DFT

```
//Example 6.13
clc;clear;close;
X=[1 2 3 4];
//Calculation of IDFT
x=fft(X,1);
```

```
x=clean(x);
disp(X,'DFT of the Sequence is X(k): ');
disp(x,'Sequence is x(n): ');
```

Scilab code Exa 6.14 Inverse DFT

```
//Example 6.14
clc;clear;close;
X=[3 2+%i 1 2-%i];
//Calculation of IDFT
x=fft(X,1);
x=clean(x);
disp(X,'DFT of the Sequence is X(k): ');
disp(x,'Sequence is x(n): ');
```

Scilab code Exa 6.15 DIT FFT

```
//Example 6.15

clc;clear;
x=[1 2 3 4 4  3 2 1];
X=clean(fft(x));
disp(x,'x(n)=');
disp(X,'X(k)=');
```

Scilab code Exa 6.16 DIT FFT

```
//Example 6.16

clc;clear;
```

```
x=[0 1 2 3 4 5 6 7];
X=clean(fft(x));
disp(x,'x(n)=');
disp(X,'X(k)=');
```

Scilab code Exa 6.17 DIT FFT

```
//Example 6.17

clc;clear;
n=0:7;
x=2^n;
X=clean(fft(x));
disp(x,'x(n)=');
disp(X,'X(k)=');
```

Scilab code Exa 6.18 DIT FFT

```
//Example 6.18

clc;clear;
x=[0 1 2 3];
X=clean(fft(x));
disp(x,'x(n)=');
disp(X,'X(k)=');
```

Scilab code Exa 6.19 DIF FFT

```
//Example 6.19

```

```
clc;clear;
x=[1 2 3 4 4 3 2 1];
X=clean(fft(x));
disp(x,'x(n)=');
disp(X,'X(k)=');
```

Scilab code Exa 6.20 DIF FFT

```
//Example 6.20

clc;clear;
n=0:7;
x=2^n;
X=clean(fft(x));
disp(x,'x(n)=');
disp(X,'X(k)=');
```

Scilab code Exa 6.21 DIF FFT

```
//Example 6.21

clc;clear;
n=0:7;
x=n+1;
X=clean(fft(x));
disp(x,'x(n)=');
disp(X,'X(k)=');
```

Scilab code Exa 6.22 DIF FFT

```
//Example 6.21

clc;clear;
n=0:3;
x=cos(n*%pi/2);
X=clean(fft(x));
disp(x,'x(n)=');
disp(X,'X(k)=');
```

Scilab code Exa 6.23 IFFT

```
//Example 6.23

clc;clear;
X=[6 -2+2*%i -2 -2-2*%i];
x=clean(ifft(X));
disp(X,'X(k)=');
disp(x,'x(n)=');
```

Scilab code Exa 6.24 IFFT

```
//Example 6.24

clc;clear;
X=[20 -5.828-2.414*%i 0 -0.172-0.414*%i 0
    -0.172+0.414*%i 0 -5.828+2.414*%i];
x=round(clean(ifft(X)));
disp(X,'X(k)=');
disp(x,'x(n)=');
```

Scilab code Exa 6.25 IFFT

```
//Example 6.25

clc;clear;
X=[255 48.63+166.05*%i -51+102*%i -78.63+46.05*%i
   -85 -78.63-46.05*%i -51-102*%i 48.63-166.05*%i];
x=round(clean(ifft(X)));
disp(X,'X(k)=');
disp(x,'x(n)=');
```

Scilab code Exa 6.26 IFFT

```
//Example 6.26

clc;clear;
X=[36 -4+9.656*%i -4+4*%i -4+1.656*%i -4 -4-1.656*%i
   -4-4*%i -4-9.656*%i ];
x=round(clean(ifft(X)));
disp(X,'X(k)=');
disp(x,'x(n)=');
```

Scilab code Exa 6.27 IFFT

```
//Example 6.27

clc;clear;
t=0:0.0025:0.0175;
f=50;
x=sin(2*%pi*f*t);
X=clean(fft(x));
disp(x,'x(n)=');
disp(X,'X(k)=');
```

Scilab code Exa 6.34 `Overlap Add Convolution`

```
//Example 6.34

clc;clear;close;
h=[2 2 1];
x=[3 0 -2 0 2 1 0 -2 -1 0];
M=length(h);                    //length of impulse response
L=2^M;                          //length of FFT/IFFT operation
N=L-M+1;
xl=length(x);
K=ceil(xl/N);                    //number of iterations
h=[h zeros(1,L-M)];
x=[x x(1:K*N-xl)];
H=fft(h);
y=zeros(1,M-1);
for k=0:K-1
    xk=[x(k*N+1:(k+1)*N) zeros(1,M-1)];
    Xk=fft(xk);
    Yk=H.*Xk;
    yk=ifft(Yk);
    yk=clean(yk);
    y=[y(1:k*N) y(k*N+1:k*N+M-1)+yk(1:M-1) yk(M:L)];
    disp(k+1,'Segment =');
    disp(xk,'xk(n)=');
    disp(yk,'yk(n)=');
end
y=y(1:xl+M-1);
disp(y,'Output Sequence is y(n): ');
```

Scilab code Exa 6.35 Overlap Save Convolution

```
//Example 6.35

clc;clear;close;
h=[2 2 1];
x=[3 0 -2 0 2 1 0 -2 -1 0];
M=length(h);                    //length of impulse
    response
L=2^M;                  //length of FFT/IFFT operation
N=L-M+1;
xl=length(x);
K=ceil(xl/N);           //number of iterations
h=[h zeros(1,L-M)];
x=[zeros(1,M-1) x x(1:K*N-xl)];
H=fft(h);
for k=0:K-1
    xk=x(k*N+1:(k+1)*N+M-1);
    Xk=fft(xk);
    Yk=H.*Xk;
    yk=ifft(Yk);
    yk=clean(yk);
    y=[yk(1:k*N) yk(M:L)];
    disp(k+1,'Segment =');
    disp(xk,'xk(n)=');
    disp(yk,'yk(n)=');
end
disp(y,'Output Sequence is y(n): ');
```

Scilab code Exa 6.36 Cross Correlation

```
//Example 6.36

clc;clear;close;
x=[1 0 0 1];
```

```
h=[4 3 2 1];
ylength=length(x)+length(h)-1;
xlength=length(x);
x=[zeros(1,length(h)-1) x zeros(1,length(h)-1)];
y=0;
//Calculation of cross correlation
for n=1:ylength;
    y(n)=x*[zeros(1,n-1) h zeros(1,ylength-n)]';
                //this instruction performs cross
       correlation of x & h
end

disp(x,'First Sequence is x(n): ');
disp(h,'Second Sequence is h(n): ');
disp(y,'Correlation Sequence y[n] is');
figure;
subplot(3,1,1);
plot2d3(x);
title('First Seqence x[n]:');ylabel('Amplitude-->');
   xlabel('n-->')
subplot(3,1,2);
plot2d3(h);
title('Second Seqence h[n]:');ylabel('Amplitude-->')
   ;xlabel('n-->')
subplot(3,1,3);
plot2d3(y);
title('Correlation Seqence y[n]:');ylabel('Amplitude
   -->');xlabel('n-->')
```

Scilab code Exa 6.37 Circular Correlation

```
//Example 6.37

clc;clear;close;
x=[1 0 0 1];
```

```
h=[4 3 2 1];
ylength=length(x);
y=0;
//Calculation of circular correlation
for n=1:ylength,
    y(n)=0;
    for k=1:ylength,
        l=k-n+1;
        if l <= 0 then
            l=l+ylength;
        end
     y(n)=y(n)+(x(k)*h(l));
     end
     y(n)=y(n)/4;
end

disp(x,'First Sequence is x(n): ');
disp(h,'Second Sequence is h(n): ');
disp(y,'Correlation Sequence y[n] is');
figure;
subplot(3,1,1);
plot2d3(x);
title('First Seqence x[n]:');ylabel('Amplitude—>');
   xlabel('n—>')
subplot(3,1,2);
plot2d3(h);
title('Second Seqence h[n]:');ylabel('Amplitude—>')
   ;xlabel('n—>')
subplot(3,1,3);
plot2d3(y);
title('Correlation Seqence y[n]:');ylabel('Amplitude
   —>');xlabel('n—>')
```

Chapter 7

Finite Impulse Response Filters

Scilab code Exa 7.3 Low pass filter using fourier series method

```
//Example 7.3

clc;clear;close;
fp=2000;          //passband frequency
F=9600;           //sampling frequancy

//Calculation of filter co-efficients
a0=1/F*integrate('1','t',-fp,fp);
for n=1:10;
a(1,n)=2/F*integrate('cos(2*%pi*n*f/F)','f',-fp,fp);
end
h=[a(:,$:-1:1)/2 a0 a/2];

//Displaying filter co-efficients
disp(F,'Sampling frequency F= ',fp,'Assumption: Passband frequency fp= ');
disp('Filter co-efficients:');
disp(a0,'h(0)= ');disp(a/2,'h(n)=h(-n)=');

n=-10:10;
plot2d3(n,h);
```

```
title('Filter transfer function h(n)');xlabel('n—>'
   );
```

Scilab code Exa 7.4 Low pass filter using Type 1 frequency sampling technique

```
//Example 7.4

clc;clear;close;
M=7;w=2*%pi/M;

//Calculation of filter co-efficients
k=[0 1 6];
for n=0:M-1
    h(n+1)=sum(exp(-%i*3*w*k).*exp(%i*w*k*n))/M;
end
h=clean(h);

//Displaying filter co-efficients
disp(M,'Filter Order M= ');
disp('Filter co-efficients:');
disp(h,'h(n)= ');

plot2d3(h);
title('Filter transfer function h(n)');xlabel('n—>'
   );
```

Chapter 8

Infinite Impulse Response Filters

Scilab code Exa 8.1 IIR filter Design byBackward Difference For Derivative method

```
//Example 8.1
clc;clear;close;
s=poly(0,'s');
z=poly(0,'z');
T=1;
Hs=1/(s+2);
Hz=horner(Hs,(1-1/z)/T);
disp('Using Backward difference formula for
    derivative:')
disp(Hs,'H(s)=');
disp(Hz,'H(z)=');
```

Scilab code Exa 8.2 IIR filter Design byBackward Difference For Derivative method

```
//Example 8.2
clc;clear;close;
```

```
s=poly(0,'s');
z=poly(0,'z');
T=1;
Hs=1/(s^2+16);
Hz=horner(Hs,(1-1/z)/T);
disp('Using Backward difference formula for derivative:')
disp(Hs,'H(s)=');
disp(Hz,'H(z)=');
```

Scilab code Exa 8.3 `IIR filter Design byBackward Difference For Derivative method`

```
//Example 8.3
clc;clear;close;
s=poly(0,'s');
z=poly(0,'z');
T=1;
Hs=1/((s+0.1)^2+9);
Hz=horner(Hs,(1-1/z)/T);
disp('Using Backward difference formula for derivative:')
disp(Hs,'H(s)=');
disp(Hz,'H(z)=');
```

Scilab code Exa 8.4 `IIR filter Design by Impulse Invariant method`

```
//Example 8.4
clc;clear;close;
s=poly(0,'s');
z=poly(0,'z');
T=1;
Hs=(s+0.2)/((s+0.2)^2+9);
Hz=horner(Hs,(1-1/z)/T);
```

```
disp('Using Impulse Invariant Technique:')
disp(Hs,'H(s)=');
disp(Hz,'H(z)=');
```

Scilab code Exa 8.5 IIR filter Design by Impulse Invariant method

```
//Example 8.5
clc;clear;close;
s=poly(0,'s');
z=poly(0,'z');
T=1;
Hs=1/(s+1)/(s+2);
Hz=horner(Hs,(1-1/z)/T);
disp('Using Impulse Invariant Technique:')
disp(Hs,'H(s)=');
disp(Hz,'H(z)=');
```

Scilab code Exa 8.6 IIR filter Design by Impulse Invariant method

```
//Example 8.6
clc;clear;close;
s=poly(0,'s');
z=poly(0,'z');
T=1;
Hs=1/(s+0.5)/(s^2+0.5*s+2);
Hz=horner(Hs,(1-1/z)/T);
disp('Using Impulse Invariant Technique:')
disp(Hs,'H(s)=');
disp(Hz,'H(z)=');
```

Scilab code Exa 8.7 `IIR filter Design by Bilinear Transformation method`

```
//Example 8.7
clc;clear;close;
s=poly(0,'s');
z=poly(0,'z');
T=0.276;
Hs=(s+0.1)/((s+0.1)^2+9);
Hz=ss2tf(cls2dls(tf2ss(Hs),T));
disp('Using Bilinear Transformation:');
disp(Hs,'H(s)=');
disp(Hz,'H(z)=');
```

Scilab code Exa 8.8 `IIR filter Design by Bilinear Transformation method`

```
//Example 8.8
clc;clear;close;
s=poly(0,'s');
z=poly(0,'z');
T=0.1;
Hs=2/(s+1)/(s+2);
Hz=ss2tf(cls2dls(tf2ss(Hs),T));
disp(Hs,'H(s)=');
disp(Hz,'H(z)=');
```

Scilab code Exa 8.9 `IIR filter Design by Bilinear Transformation method`

```
//Example 8.9
clc;clear;close;
s=poly(0,'s');
z=poly(0,'z');
T=0.1;
wr=0.25*%pi;          //Given cut off frequency
```

```
fc=2/T*tan(wr/2);
Hs=fc/(s+fc);
Hz=ss2tf(cls2dls(tf2ss(Hs),T));
disp('Using Bilinear Transformation:');
disp(Hs,'H(s)=');
disp(Hz,'H(z)=');
```

Scilab code Exa 8.10 IIR filter Design by Bilinear Transformation method

```
//Example 8.10
clc;clear;close;
s=poly(0,'s');
z=poly(0,'z');
T=0.1;
Hs=1/(s+1)^2;
Hz=ss2tf(cls2dls(tf2ss(Hs),T));
disp('Using Bilinear Transformation:');
disp(Hs,'H(s)=');
disp(Hz,'H(z)=');
```

Scilab code Exa 8.11 Butterworth Filter using Impulse Invariant transformation

```
//Example 8.11
clc;clear;close;
rp=0.707                          //passband ripple
rs=0.2                           //stopband ripple
wp=%pi/2;                        //passband frequency
ws=3*%pi/4;                      //stopband frequency
T=1;
fp=wp/T;
fs=ws/T;
s=poly(0,'s');
z=poly(0,'z');
```

```
hs=1;
//Calculating the order of filter
num=log((rs^-2 -1)/(rp^-2 -1));
den=2*log(fs/fp);
N=ceil(num/den);

//Calculation of cut-off frequency
fc=fp/(rp^-2 -1)^(0.5/N);

//Calculating filter response
if modulo(N,2)==1 then
    b=-2*sin(%pi/(2*N));
    hs=hs*fc/(s+fc);
end
for k=1:N/2
    b=2*sin((2*k-1)*%pi/(2*N));
    hs=hs*fc^2/(s^2+b*fc*s+fc^2);
end
hs=clean(hs);
sys=syslin('c',hs);
hz=horner(ss2tf(dscr(sys,T)),1/z);          //converting H(s) to H(z)

//Displaying filter response
[hzm,fr]=frmag(hz,256);
disp(hz,'Filter Transfer function: ');
plot(fr,hzm);
title('Lowpass Butterworth Filter Response');ylabel('Amplitude-->');xlabel('Normalised frequency f/fs -->');
```

Scilab code Exa 8.12 Butterworth Filter using Bilinear transformation

```
//Example 8.12
clc;clear;close;
```

```
rp=0.9                                  //passband ripple
rs=0.2                                  //stopband ripple
wp=%pi/2;                               //passband frequency
ws=3*%pi/4;                             //stopband frequency
T=1;
fp=2/T*tan(wp/2);
fs=2/T*tan(ws/2);
s=poly(0,'s');
z=poly(0,'z');
hs=1;
//Calculating the order of filter
num=log((rs^-2 -1)/(rp^-2 -1));
den=2*log(fs/fp);
N=ceil(num/den);

//Calculation of cut-off frequency
fc=fp/(rp^-2 -1)^(0.5/N);

//Calculating filter response
if modulo(N,2)==1 then
    hs=hs*fc/(s+fc);
end
for k=1:N/2
    b=2*sin((2*k-1)*%pi/(2*N));
    hs=hs*fc^2/(s^2+b*fc*s+fc^2);
end
hs=clean(hs);
sys=syslin('c',hs);
hz=ss2tf(cls2dls(tf2ss(sys),T));          //converting H(s) to H(z)

//Displaying filter response
[hzm,fr]=frmag(hz,256);
disp(hz,'Filter Transfer function: ');
plot(fr,hzm);
title('Lowpass Butterworth Filter Response');ylabel('Amplitude-->');xlabel('Normalised frequency f/fs-->');
```

Scilab code Exa 8.14 Filter transformation

```
//Example 8.14
clc;clear;close;
s=poly(0,'s');
fc=1;                //Assumed cut off frequency
Q=10;f0=2;          //Given data
Hs=1/(s^2+2*s+1);
l=fc*(s^2+f0^2)/(s*f0/Q);
Hs1=horner(Hs,l);
disp(Hs,'Low pass filter H(s)=');
disp(Hs1,'Band pass filterH(s)=');
```

Scilab code Exa 8.15 Filter transformation

```
//Example 8.15
clc;clear;close;
s=poly(0,'s');
fc=1;                //Assumed cut off frequency of
   low pass filter
f0=5;               //Assumed cut off frequency of
   high pass filter
Hs=fc/(s+fc);
Hs1=horner(Hs,fc*f0/s);
disp(Hs,'H(s)=',fc,'Low pass filter with fc=');
disp(Hs1,'H(s)=',f0,'High pass filter with fc=');
```

Chapter 9

Realisation of Digital Linear Systems

Scilab code Exa 9.4 Cascade Realisation

```
//Example 9.4
clc;clear;close;
z=poly(0,'z');
Hz=2*(z+2)/(z*(z-0.1)*(z+0.5)*(z+0.4));
H=dscr(Hz,0.1);
disp(Hz,'System Function H(z)=');
disp(H,'System Function for cascade realisation Hk(z
    )=');
```

Scilab code Exa 9.5.a Parallel Realisation

```
//Example 9.5.a
clc;clear;close;
z=poly(0,'z');
s=poly(0,'s');
Hz=3*(2*z^2+5*z+4)/(2*z+1)/(z+2);
```

```
H=pfss(Hz/z);
for k=1:length(H)
    H(k)=clean(H(k));
    H1(k)=z*horner(H(k),z);
disp(H1(k),'System Function for parallel realisation
    Hk(z)=');
end
disp(Hz,'System Function H(z)=');
```

Scilab code Exa 9.5.b Parallel Realisation

```
//Example 9.5.b
clc;clear;close;
z=poly(0,'z');
s=poly(0,'s');
Hz=3*z*(5*z-2)/(z+1/2)/(3*z-1);
H=pfss(Hz/z);
for k=1:length(H)
    H(k)=clean(H(k));
    H1(k)=z*horner(H(k),z);
disp(H1(k),'System Function for parallel realisation
    Hk(z)=');
end
disp(Hz,'System Function H(z)=');
```

Chapter 10

Effects of Finite Word Length in Digital Filters

Scilab code Exa 10.2 `Output Quantisation Noise`

```
//Example 10.2

clc;clear;close;
z=poly(0,'z');
H=0.5*z/(z-0.5);
B=8;
pn=2^(-2*B)/12;          //Noise power
X=H*horner(H,1/z)/z;
r=roots(denom(X));
rl=length(r);
rc=coeff(denom(X))
q1=[];q2=[];
for n=1:rl                //Loop to separate poles
   inside the unit circle
    if (abs(r(n))<1) then
        q1=[q1 r(n)];
    else
        q2=[q2 r(n)];
    end
```

```
end
P=numer(X)/rc(length(rc));
Q1=poly(q1,'z');
Q2=poly(q2,'z');
I=residu(P,Q1,Q2);          //Residue Calculation
po=pn*I;                    //Output Noise power
disp(pn,'Input Noise power');
disp(po,'Output Noise power');
```

Scilab code Exa 10.3 Deadband Interval

```
//Example 10.3

clc; clear;
//y(n)=0.9y(n-1)+x(n)
//Input x(n)=0
n=-1;y=12;          //Initial Condition y(-1)=12
flag=1;
while n<8
    n=n+1;
    y=[y 0.9*y(n+1)];
    yr=round(y);
end
disp(n,'n=');
disp(y,'y(n)-exact');
disp(yr,'y(n)-rounded');
disp([-yr(n+2) yr(n+2)],'Deadband interval ')
```

Scilab code Exa 10.4 Deadband Interval

```
//Example 10.4

clc; clear;
```

```
//y(n)=0.9y(n-1)+x(n)
a=0.9;
l=ceil(0.5/(1-abs(a)));
disp([-l l],'Deadband interval ')
```

Scilab code Exa 10.5 `Output Quantisation Noise for Cascade realisation`

```
//Example 10.5

clc;clear;close;
x=poly(0,'x');          //x=2^-2B
z=poly(0,'z');
H1=1/(1-0.9/z);
H2=1/(1-0.8/z);
H=H1*H2;
pn=x/12;                //Input Noise power

//Calculation of output noise for H1(z)
X1=H*horner(H,1/z)/z;
r1=roots(denom(X1));
rc1=coeff(denom(X1));
q1=[];s1=[];
for n=1:length(r1)              //Loop to separate
    poles inside the unit circle
    if (abs(r1(n))<1) then
        q1=[q1 r1(n)];
    else
        s1=[s1 r1(n)];
    end
end
P1=numer(X1)/rc1(length(rc1));
Q1=poly(q1,'z');
S1=poly(s1,'z');
I1=abs(residu(P1,Q1,S1));       //Residue
   Calculation
```

```
po1=pn*I1;                    //Output Noise power

//Calculation of output noise for H2(z)
X2=H2*horner(H2,1/z)/z;
r2=roots(denom(X2));
rc2=coeff(denom(X2));
q2=[];s2=[];
for n=1:length(r2)                //Loop to separate poles inside the unit circle
    if (abs(r2(n))<1) then
        q2=[q2 r2(n)];
    else
        s2=[s2 r2(n)];
    end
end
P2=numer(X2)/rc2(length(rc2));
Q2=poly(q2,'z');
S2=poly(s2,'z');
I2=abs(residu(P2,Q2,S2));         //Residue Calculation
po2=pn*I2;                        //Output Noise power

po=po1+po2;
disp(pn,'Input Noise power');
disp(I1,'I1=');disp(I2,'I2=');
disp(po1,'Output Noise power for H1(z)');
disp(po2,'Output Noise power for H2(z)');
disp(po,'Total Output Noise power');
```

Chapter 11

Multirate Digital Signal Processing

Scilab code Exa 11.1 `Time Decimation`

```
//Example 11.1

clc;clear;close;
x=[0:6 0:6];
y=x(1:3:length(x));
disp(x,'Input signal x(n)=');
disp(y,'Output signal of decimation process by
   factor three y(n)');
subplot(2,1,1);
plot2d3(x);title('Input signal x(n)');
subplot(2,1,2);
plot2d3(y);title('Output signal y(n)');
```

Scilab code Exa 11.2 `Interpolation`

```
//Example 11.2
```

```

clc;clear;close;
x=0:5;
y=[];
for i=1:length(x)
    y(1,2*i)=x(i);
end
disp(x,'Input signal x(n)=');
disp(y,'Output signal of interpolation process with factor two y(n)');
subplot(2,1,1);
plot2d3(x);title('Input signal x(n)');
subplot(2,1,2);
plot2d3(y);title('Output signal y(n)');
```

Scilab code Exa 11.4 Polyphase Decomposition

```
//Example 11.4

clc;clear;
z=poly(0,'z');
num=1-4*z^-1;
den=1+5*z^-1;
H=num/den;
num1=num*(1-5*z^-1);
den1=den*(1-5*z^-1);
H1=num1/den1;
c=coeff(numer(num1));
clength=length(c);
c=[c zeros(1,pmodulo(clength,2))];       //make length of 'c' multiple of 2
c0=[];c1=[];
for n=1:ceil(clength/2)                  //loop to separate even and odd powers of z
    c0=[c0 c(2*n-1) 0];
```

```
    c1=[c1 c(2*n) 0];
end
E0=poly(c0,'z','coeff')/z^n/den1;
E1=poly(c1,'z','coeff')/z^(n-2)/den1;
disp('Polyphase Components')
disp(E0,'E0(z)');
disp(E1,'E1(z)');
```

Scilab code Exa 11.5 Decimator implementation

```
//Example 11.5

clc;clear;

function [N,R]=func(Fs,Fp,Ft,Fti,dp,ds,M)
    dF=(Fs-Fp)/Ft;                                  //Normalised transition bandwidth
    N=round((-20*log10(sqroot(dp*ds))-13)/(14.6*dF));   //FIR Filter length
    R=N*Fti/M;                                      //Number of Multiplications per second
endfunction

Ft=20000;           //Sampling rate of input signal
Fp=40;              //Passband frequency
Fs=50;              //Stopband frequency
dp=0.01;            //Passband ripple
ds=0.002;           //Stopband ripple
M=100;              //Decimation Factor
Fti=Ft;             //Input sampling rate
//Single stage implementation
[N1,R1]=func(Fs,Fp,Ft,Fti,dp,ds,M);

```

```
//Two stage implementation
//Stage 1 F(z) with decimation factor 50
Fpf=Fp;              //Passband frequency
Fsf=190;             //Stopband frequency
dpf=0.005;           //Passband ripple
dsf=0.002;           //Stopband ripple
Mf=50;               //Decimation Factor
Fti=Ft;              //Input sampling rate
[N2f,R2f]=func(Fsf,Fpf,Ft,Fti,dpf,dsf,Mf);

//Stage 2 G(z) with decimation factor 2
Fpg=50*Fp;           //Passband frequency
Fsg=50*Fs;           //Stopband frequency
dpg=0.005;           //Passband ripple
dsg=0.002;           //Stopband ripple
Mg=2;                //Decimation Factor
Fti=Ft/50;           //Input sampling rate
[N2g,R2g]=func(Fsg,Fpg,Ft,Fti,dpg,dsg,Mg);
N2=N2f+50*N2g+2;      //Total filter length
R2=R2f+R2g;           //Total Number of
   Multiplications per second
disp(R1,'Number of Multiplications per second =',N1,
   'FIR filter length =','For Single stage
   implementation:');
disp('For Two stage implementation:');
disp(R2f,'Number of Multiplications per second =',
   N2f,'FIR filter length =','For F(z):');
disp(R2g,'Number of Multiplications per second =',
   N2g,'FIR filter length =','For G(z):');
disp(R2,'Total Number of Multiplications per second
   =',N2,'Overall FIR filter length =');
```

Scilab code Exa 11.6 Decimator implementation

```
//Example 11.6
```

```

clc;clear;


function [N,R]=func(Fs,Fp,Ft,Fti,dp,ds,M)
    dF=(Fs-Fp)/Ft;                                              //
       Normalised transition bandwidth
    N=round((-20*log10(sqroot(dp*ds))-13)/(14.6*dF))
       ;   //FIR Filter length
    R=N*Fti/M;                                                      //
       Number of Multiplications per second
endfunction

Ft=10000;         //Sampling rate of input signal
Fp=150;            //Passband frequency
Fs=180;            //Stopband frequency
dp=0.002;          //Passband ripple
ds=0.001;         //Stopband ripple
M=20;            //Decimation Factor
Fti=Ft;           //Input sampling rate
//Single stage implementation
[N1,R1]=func(Fs,Fp,Ft,Fti,dp,ds,M);

//Two stage implementation
//Stage 1 F(z) with decimation factor 50
Fpf=Fp;             //Passband frequency
Fsf=720;            //Stopband frequency
dpf=0.001;          //Passband ripple
dsf=0.001;          //Stopband ripple
Mf=10;              //Decimation Factor
Fti=Ft;             //Input sampling rate
[N2f,R2f]=func(Fsf,Fpf,Ft,Fti,dpf,dsf,Mf);

//Stage 2 G(z) with decimation factor 2
Fpg=10*Fp;          //Passband frequency
Fsg=10*Fs;          //Stopband frequency
```

```
dpg=0.001;           //Passband ripple
dsg=0.001;           //Stopband ripple
Mg=2;                //Decimation Factor
Fti=Ft/10;           //Input sampling rate
[N2g,R2g]=func(Fsg,Fpg,Ft,Fti,dpg,dsg,Mg);
N2=N2f+10*N2g+2;      //Total filter length
R2=R2f+R2g;           //Total Number of Multiplications per second

disp(R1,'Number of Multiplications per second =',N1,'FIR filter length =','For Single stage implementation:');
disp('For Two stage implementation:');
disp(R2f,'Number of Multiplications per second =',N2f,'FIR filter length =','For F(z):');
disp(R2g,'Number of Multiplications per second =',N2g,'FIR filter length =','For G(z):');
disp(R2,'Total Number of Multiplications per second =',N2,'Overall FIR filter length =');
```

Chapter 12

Spectral Estimation

Scilab code Exa 12.2 Power Spectrum

```
//Example 12.2
clc;clear;close;
N=8;n=0:N-1;
f1=0.6;f2=0.62;
x=cos(2*%pi*f1*n)+cos(2*%pi*f2*n);
L1=8;
for k=0:L1-1
    P1(k+1)=1/N*abs(x*(cos(%pi*n*k/L1)-%i*sin(%pi*n*
       k/L1))')^2
end
L2=16;
for k=0:L2-1
    P2(k+1)=1/N*abs(x*(cos(%pi*n*k/L2)-%i*sin(%pi*n*
       k/L2))')^2;
end
L3=32;
for k=0:L3-1
    P3(k+1)=1/N*abs(x*(cos(%pi*n*k/L3)-%i*sin(%pi*n*
       k/L3))')^2;
end
subplot(311);
```

```
plot2d3(0:L1-1,P1);title('L=8');
subplot(312);
plot2d3(0:L2-1,P2);title('L=16');
subplot(313);
plot2d3(0:L3-1,P3);title('L=32');
```

Scilab code Exa 12.4 Frequency resolution

```
//Example 12.4
clc;clear;close;
N=1000;
Q=10;
disp(N,'Length of sample sequence N=',Q,'Quality
    factor Q=');
f_bart=Q/(1.11*N);
f_w=Q/(1.39*N);
f_bt=Q/(2.34*N);
disp(f_bart,'Bartlett Frequency resolution =');
disp(f_w,'Welch Frequency resolution =');
disp(f_bt,'Blackman Turkey Frequency resolution =');
```