Scilab Textbook Companion for Signals And Systems by A. V. Oppenheim, A. S. Willsky And S. H. Nawab[1]

Created by
Prof. R. Senthilkumar
B.Tech
Electrical Engineering
Institute of Road and Transport Technology
College Teacher
Na
Cross-Checked by
Prof. Saravanan Vijayakumaran

July 9, 2014

[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

Book Description

Title: Signals And Systems

Title: Signals And Systems

Author: A. V. Oppenheim, A. S. Willsky And S. H. Nawab

Author: A. V. Oppenheim, A. S. Willsky And S. H. Nawab

Publisher: PHI Learning, New Delhi

Publisher: PHI Learning, New Delhi

Edition: 2

Edition: 2

Year: 1992

Year: 1992

ISBN: 978-81-203-1246-3

ISBN: 978-81-203-1246-3

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

Scilab numbering policy used in this document and the relation to the above book.Example (Solved example) Example (Solved example) Equation (Particular equation of the above book) Equation (Particular equation of the above book) 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

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

Signals and Systems

Chapter 2

Signals and Systems

Scilab code Exa 1.1 Time Shifting

```
//clear//
//Example 1.1: Time Shifting
//SIGNALS & SYSTEMS, Second Edition
//V.OPPENHEIM, S.WILLSKY, S.HAMID NAMWAB
//PHI, 2008 Edition
//Page 10
clear;
clc;
close;
t = 0:1/100:1;
for i = 1:length(t)
  x(i) = 1 ;
end
for i = length(t)+1:2*length(t)
  x(i) = 1-t(i-length(t));
end
t1 = 0:1/100:2;
t2 = -1:1/100:1;
//t3 = 0:1/100:4/3;
```

```
//t4 = 0:1/length(t3):1;
//Mid =ceil(length(t3)/2);
//for i = 1:Mid
//   x3(i) = 1 ;
//end
//for i = Mid+1:length(t3)
//   x3(i) = 1-t4(i-Mid);
//end
figure
a=gca();
plot2d(t1,x(1:$-1))
a.thickness=2;
xtitle('The signal x(t)')
figure
a=gca();
plot2d(t2,x(1:$-1))
a.thickness=2;
a.y_location = "middle";
xtitle('The signal x(t+1)')
figure
a=gca();
plot2d(t2,x($:-1:2))
a.thickness=2;
a.y_location = "middle";
xtitle('The signal x(-t+1)')
```

Scilab code Exa 1.1

Time Shifting

```
//clear//
//Example 1.1: Time Shifting
//SIGNALS & SYSTEMS, Second Edition
//V.OPPENHEIM, S.WILLSKY, S.HAMID NAMWAB
//PHI, 2008 Edition
//Page 10
clear;
clc;
```

```
close;
t = 0:1/100:1;
for i = 1:length(t)
  x(i) = 1 ;
end
for i = length(t)+1:2*length(t)
  x(i) = 1-t(i-length(t));
end
t1 = 0:1/100:2;
t2 = -1:1/100:1;
//t3 = 0:1/100:4/3;
//t4 = 0:1/length(t3):1;
//Mid =ceil(length(t3)/2);
//for i = 1:Mid
//  x3(i) = 1 ;
//end
//for i = Mid+1:length(t3)
//  x3(i) = 1-t4(i-Mid);
//end
figure
a=gca();
plot2d(t1,x(1:$-1))
a.thickness=2;
xtitle('The signal x(t)')
figure
a=gca();
plot2d(t2,x(1:$-1))
a.thickness=2;
a.y_location = "middle";
xtitle('The signal x(t+1)')
figure
a=gca();
plot2d(t2,x($:-1:2))
a.thickness=2;
a.y_location = "middle";
xtitle('The signal x(-t+1)')
```

Scilab code Exa 1.2 Time Scaling

```
//clear//
//Example 1.2:Time Scaling
//SIGNALS & SYSTEMS, Second Edition
//V.OPPENHEIM, S.WILLSKY, S.HAMID NAMWAB
//PHI, 2008 Edition
//Page 11
clear;
clc;
close;
t3 = 0:1/100:4/3;
t4 = 0:1/length(t3):1;
Mid =ceil(length(t3)/2);
for i = 1:Mid
  x3(i) = 1 ;
end
for i = Mid+1:length(t3)
  x3(i) = 1-t4(i-Mid);
end
figure
a=gca();
plot2d(t3,x3)
a.thickness=2;
xtitle('Time Scaling x(3t/2)')
```

Scilab code Exa 1.2

Time Scaling

```
//clear//
//Example 1.2:Time Scaling
//SIGNALS & SYSTEMS, Second Edition
```

```
//V.OPPENHEIM, S.WILLSKY, S.HAMID NAMWAB
//PHI, 2008 Edition
//Page 11
clear;
clc;
close;
t3 = 0:1/100:4/3;
t4 = 0:1/length(t3):1;
Mid =ceil(length(t3)/2);
for i = 1:Mid
  x3(i) = 1 ;
end
for i = Mid+1:length(t3)
  x3(i) = 1-t4(i-Mid);
end
figure
a=gca();
plot2d(t3,x3)
a.thickness=2;
xtitle('Time Scaling x(3t/2)')
```

Scilab code Exa 1.3 Time Scaling and Time Shifting

```
//clear//
//Example 1.3:Time Scaling and Time Shifting
//SIGNALS & SYSTEMS, Second Edition
//V.OPPENHEIM, S.WILLSKY, S.HAMID NAMWAB
//PHI, 2008 Edition
//Page 11
clear;
clc;
close;
t3 = 0:1/100:4/3;
```

```
t4 = 0:1/length(t3):1;
Mid =ceil(length(t3)/2);
for i = 1:Mid
  x3(i) = 1 ;
end
for i = Mid+1:length(t3)
  x3(i) = 1-t4(i-Mid);
end
t5 = -2/3:1/100:2/3;
figure
a=gca();
plot2d(t5,x3)
a.thickness=2;
a.y_location ="middle";
xtitle('Time Scaling and Time Shifting x((3t/2)+1)')
```

Scilab code Exa 1.3

Time Scaling and Time Shifting

```
//clear//
//Example 1.3:Time Scaling and Time Shifting
//SIGNALS & SYSTEMS, Second Edition
//V.OPPENHEIM, S.WILLSKY, S.HAMID NAMWAB
//PHI, 2008 Edition
//Page 11
clear;
clc;
close;
t3 = 0:1/100:4/3;
t4 = 0:1/length(t3):1;
Mid =ceil(length(t3)/2);
for i = 1:Mid
  x3(i) = 1 ;
end
for i = Mid+1:length(t3)
  x3(i) = 1-t4(i-Mid);
end
```

```
t5 = -2/3:1/100:2/3;
figure
a=gca();
plot2d(t5,x3)
a.thickness=2;
a.y_location ="middle";
xtitle('Time Scaling and Time Shifting x((3t/2)+1)')
```

Scilab code Exa 1.4 Combinationation two periodic signals

```
//clear//
//Example 1.4:Combinationation two periodic signals
// Aperiodic signal
//Page 12
clear;
clc;
close;
F=1;   //Frequency  = 1 Hz
t1 = 0:-1/100:-2*%pi;
x1 = cos(F*t1);
t2 = 0:1/100:2*%pi;
x2 = sin(F*t2);
a=gca();
plot(t2,x2);
plot(t1,x1);
a.y_location = "middle";
a.x_location = "middle";
xtitle('The signal x(t) = cost for t < 0 and sint
    for t > 0: Aperiodic Signal')
```

Scilab code Exa 1.4

Combinationation two periodic signals

```
//clear//
//Example 1.4:Combinationation two periodic signals
// Aperiodic signal
//Page 12
clear;
clc;
close;
F=1;  //Frequency  = 1 Hz
t1 = 0:-1/100:-2*%pi;
x1 = cos(F*t1);
t2 = 0:1/100:2*%pi;
x2 = sin(F*t2);
a=gca();
plot(t2,x2);
plot(t1,x1);
a.y_location = "middle";
a.x_location = "middle";
xtitle('The signal x(t) = cost for t < 0 and sint
   for t > 0: Aperiodic Signal')
```

Scilab code Exa 1.6 Fundamental period of composite discrete time signal

```
//clear//
//Example 1.6:Determine the fundamental period of
   composite
// discrete time signal
//x[n] = exp(j(2*%pi/3)n)+exp(j(3*%pi/4)n)
clear;
clc;
close;
Omega1 = 2*%pi/3;   //Angular frequency signal 1
Omega2 = 3*%pi/4;  //Angular frequency signal 2
```

```
N1 = (2*%pi)/Omega1;  //Peirod of signal 1
N2 = (2*%pi)/Omega2;  //Period of signal 2
//To find rational period of signal 1
for m1 = 1:100
  period = N1*m1;
  if(modulo(period,1)==0)
    period1 = period;
    integer_value = m1
    break;
  end
end
//To find rational period of signal 2
for m2 = 1:100
  period = N2*m2;
  if(modulo(period,1)==0)
    period2 = period;
    integer_value = m2
    break;
  end
end
disp(period1)
disp(period2)
//To determine the fundamental period N
N = period1*period2
```

Scilab code Exa 1.6

Fundamental period of composite discrete time signal

```
//clear//
//Example 1.6:Determine the fundamental period of
   composite
// discrete time signal
//x[n] = exp(j(2*%pi/3)n)+exp(j(3*%pi/4)n)
clear;
clc;
close;
Omega1 = 2*%pi/3;   //Angular frequency signal 1
```

```
Omega2 = 3*%pi/4;  //Angular frequency signal 2
N1 = (2*%pi)/Omega1;  //Peirod of signal 1
N2 = (2*%pi)/Omega2;  //Period of signal 2
//To find rational period of signal 1
for m1 = 1:100
  period = N1*m1;
  if(modulo(period,1)==0)
    period1 = period;
    integer_value = m1
    break;
  end
end
//To find rational period of signal 2
for m2 = 1:100
  period = N2*m2;
  if(modulo(period,1)==0)
    period2 = period;
    integer_value = m2
    break;
  end
end
disp(period1)
disp(period2)
//To determine the fundamental period N
N = period1*period2
```

Scilab code Exa 1.12 Classification of system

```
//clear//
//Example 1.12:Classification of system:Causality
   property
//Page 47
//To check whether the given discrete system is a
```

```
    Causal System (or) Non−Causal System
//Given discrete system y[n]= x[−n]
clear;
clc;
x = [2,4,6,8,10,0,0,0,1];  //Assign some value to
    input
n = -length(x)/2:length(x)/2;
count = 0;
mid = ceil(length(x)/2);
y = zeros(1,length(x));
y(mid+1:$) = x($:-1:mid+1);
for n = -1:-1:-mid
  y(n+1+mid) = x(-n);
end
for i = 1:length(x)
  if (y(i)==x(i))
    count = count+1;
  end
end
if (count==length(x))
    disp('The given system is a causal system')
else
    disp('Since it depends on future input value')
    disp('The given system is a non−causal system')
end
```

Scilab code Exa 1.12

Classification of system

```
//clear//
//Example 1.12:Classification of system:Causality
    property
//Page 47
//To check whether the given discrete system is a
    Causal System (or) Non−Causal System
//Given discrete system y[n]= x[−n]
clear;
```

```
clc;
x = [2,4,6,8,10,0,0,0,1];  //Assign some value to
    input
n = -length(x)/2:length(x)/2;
count = 0;
mid = ceil(length(x)/2);
y = zeros(1,length(x));
y(mid+1:$) = x($:-1:mid+1);
for n = -1:-1:-mid
  y(n+1+mid) = x(-n);
end
for i = 1:length(x)
  if (y(i)==x(i))
    count = count+1;
  end
end
if (count==length(x))
    disp('The given system is a causal system')
else
    disp('Since it depends on future input value')
    disp('The given system is a non-causal system')
end
```

Scilab code Exa 01.13 Determination of stability of a given system

```
//clear//
//Example 1.13(b):Determination of stability of a
    given system
//Page 50
//given system y(t) = exp(x(t))
clear;
clc;
Maximum_Limit = 10;
```

```
S = 0;
for t = 0:Maximum_Limit-1
  x(t+1)= -2^t;           //Input some bounded value
  S = S+exp(x(t+1));
end
if (S >Maximum_Limit)
  disp('Eventhough input is bounded output is
     unbounded')
  disp('The given system is unstable');
  disp('S =');
  S
 else
  disp('The given system is stable');
  disp(S);
end
```

Scilab code Exa 01.13

Determination of stability of a given system

```
//clear//
//Example 1.13(b):Determination of stability of a
   given system
//Page 50
//given system y(t) = exp(x(t))
clear;
clc;
Maximum_Limit = 10;
S = 0;
for t = 0:Maximum_Limit-1
  x(t+1)= -2^t;           //Input some bounded value
  S = S+exp(x(t+1));
end
if (S >Maximum_Limit)
  disp('Eventhough input is bounded output is
     unbounded')
  disp('The given system is unstable');
  disp('S =');
  S
 else
  disp('The given system is stable');
  disp(S);
end
```

Scilab code Exa 1.13 Determination of stablility of a given system

Scilab code Exa 1.13
```
//clear//
//Example 1.13:Determination of stablility of a
   given system
//Page 49
//given system y(t) = t.x(t)
clear;
clc;
x = [1,2,3,4,0,2,1,3,5,8]; //Assign some input
Maximum_Limit = 10;
S = 0;
for t = 0:Maximum_Limit-1
  S = S+t*x(t+1);
end
if (S >Maximum_Limit)
  disp('Eventhough input is bounded output is
     unbounded')
  disp('The given system is unstable');
  disp('S =');
  S
 else
  disp('The given system is stable');
  disp('The value of S =');
  S
end
```

Determination of stablility of a given system

```
//clear//
//Example 1.13:Determination of stablility of a
    given system
//Page 49
//given system y(t) = t.x(t)
clear;
clc;
x = [1,2,3,4,0,2,1,3,5,8]; //Assign some input
Maximum_Limit = 10;
S = 0;
for t = 0:Maximum_Limit-1
  S = S+t*x(t+1);
end
if (S >Maximum_Limit)
  disp('Eventhough input is bounded output is
     unbounded')
  disp('The given system is unstable');
  disp('S =');
  S
 else
  disp('The given system is stable');
  disp('The value of S =');
  S
end
```

Scilab code Exa 1.14 Time Invariance Property

```
//clear//
//Example 1.14:classification of a system:Time
    Invariance Property
//Page 51
```

```
//To check whether the given system is a Time
   variant (or) Time In-variant
// The given discrete signal is y(t) = sin(x(t))
clear;
clc;
to = 2; //Assume the amount of time shift  =2
T = 10; //Length of given  signal
for t = 1:T
  x(t) = (2*%pi/T)*t;
  y(t) = sin(x(t));
end
//First shift the input signal only
Input_shift = sin(x(T-to));
Output_shift = y(T-to);
if(Input_shift == Output_shift)
  disp('The given discrete system is a Time In-
     variant system');
else
  disp('The given discrete system is a Time Variant
     system');
end
```

Scilab code Exa 1.14

Time Invariance Property

```
//clear//
//Example 1.14:classification of a system:Time
   Invariance Property
//Page 51
//To check whether the given system is a Time
   variant (or) Time In-variant
// The given discrete signal is y(t) = sin(x(t))
clear;
clc;
to = 2; //Assume the amount of time shift  =2
T = 10; //Length of given  signal
for t = 1:T
```

```
  x(t) = (2*%pi/T)*t;
  y(t) = sin(x(t));
end
//First shift the input signal only
Input_shift = sin(x(T-to));
Output_shift = y(T-to);
if(Input_shift == Output_shift)
  disp('The given discrete system is a Time In-
     variant system');
else
  disp('The given discrete system is a Time Variant
     system');
end
```

Scilab code Exa 0.15 Sum of two complex exponentials

```
//clear//
//Example 1.5:To express sum of two complex
   exponentials
//as a single sinusoid
clear;
clc;
close;
t =0:1/100:2*%pi;
x1 = exp(sqrt(-1)*2*t);
x2 = exp(sqrt(-1)*3*t);
x = x1+x2;
for i = 1:length(x)
  X(i) = sqrt((real(x(i)).^2)+(imag(x(i)).^2));
end
plot(t,X);
xtitle('Full wave rectified sinusoid','time t','
   Magnitude');
```

Scilab code Exa 0.15

Sum of two complex exponentials

```
//clear//
//Example 1.5:To express sum of two complex
    exponentials
//as a single sinusoid
clear;
clc;
close;
t =0:1/100:2*%pi;
x1 = exp(sqrt(-1)*2*t);
x2 = exp(sqrt(-1)*3*t);
x = x1+x2;
for i = 1:length(x)
  X(i) = sqrt((real(x(i)).^2)+(imag(x(i)).^2));
end
plot(t,X);
xtitle('Full wave rectified sinusoid','time t','
    Magnitude');
```

Scilab code Exa 1.15 Classification of a System

```
//clear//
//Example 1.15:Classification of a System:Time
    Invariance Property
//Page 51
//To check whether the given system is a Time
    variant (or) Time In-variant
// The given discrete signal is y[n] = n.x[n]
clear;
clc;
```

```
no = 2; //Assume the amount of time shift  =2
L = 10; //Length of given  signal
for n = 1:L
  x(n) = n;
  y(n) = n*x(n);
end
//First shift the input signal only
Input_shift = x(L-no);
Output_shift = y(L-no);
if(Input_shift == Output_shift)
  disp('The given discrete system is a Time In-
     variant system');
else
  disp('The given discrete system is a Time Variant
     system');
end
```

Scilab code Exa 1.15

Classification of a System

```
//clear//
//Example 1.15:Classification of a System:Time
   Invariance Property
//Page 51
//To check whether the given system is a Time
   variant (or) Time In-variant
// The given discrete signal is y[n] = n.x[n]
clear;
clc;
no = 2; //Assume the amount of time shift  =2
L = 10; //Length of given  signal
for n = 1:L
  x(n) = n;
  y(n) = n*x(n);
end
//First shift the input signal only
Input_shift = x(L-no);
```

```
Output_shift = y(L-no);
if(Input_shift == Output_shift)
  disp('The given discrete system is a Time In-
     variant system');
else
  disp('The given discrete system is a Time Variant
     system');
end
```

Scilab code Exa 1.16 Time Invariance Property

```
//clear//
//Example 1.16:Classification of system:Time
   Invariance Property
//Page 52
//To check whether the given system is a Time
   variant (or) Time In-variant
// The given discrete signal is y(t) = x(2t)
clear;
clc;
to = 2; //Assume the amount of time shift  =2
T = 10; //Length of given  signal
x = [1,2,3,4,5,6,7,8,9,10];
y = zeros(1,length(x));
for t = 1:length(x)/2
    y(t) = x(2*t);
end
//First shift the input signal only
Input_shift = x(T-to);
Output_shift = y(T-to);
if(Input_shift == Output_shift)
  disp('The given discrete system is a Time In-
     variant system');
```

```
else
  disp('The given discrete system is a Time Variant
     system');
end
```

Scilab code Exa 1.16

Time Invariance Property

```
//clear//
//Example 1.16:Classification of system:Time
   Invariance Property
//Page 52
//To check whether the given system is a Time
   variant (or) Time In−variant
// The given discrete signal is y(t) = x(2t)
clear;
clc;
to = 2; //Assume the amount of time shift  =2
T = 10; //Length of given  signal
x = [1,2,3,4,5,6,7,8,9,10];
y = zeros(1,length(x));
for t = 1:length(x)/2
    y(t) = x(2*t);
end
//First shift the input signal only
Input_shift = x(T-to);
Output_shift = y(T-to);
if(Input_shift == Output_shift)
  disp('The given discrete system is a Time In−
     variant system');
else
  disp('The given discrete system is a Time Variant
     system');
end
```

Scilab code Exa 1.17 Linearity Property

```
//clear//
//Example 1.17:Classification of system:Linearity
   Property
//Page 54
//To check whether the given discrete system is a
   Linear System (or) Non-Linear System
//Given discrete system y(t)= t*x(t)
clear;
clc;
x1 = [1,1,1,1];
x2 = [2,2,2,2];
a = 1;
b = 1;
for t = 1:length(x1)
  x3(t) = a*x1(t)+b*x2(t);
end
for t = 1:length(x1)
  y1(t) = t*x1(t);
  y2(t) = t*x2(t);
  y3(t) = t*x3(t);
end
for t = 1:length(y1)
  z(t) = a*y1(t)+b*y2(t);
end
count = 0;
for n =1:length(y1)
  if(y3(t)== z(t))
    count = count+1;
  end
end
```

```
if(count == length(y3))
   disp('Since It satisifies the superposition
      principle')
   disp('The given system is a Linear system')
   y3
   z
  else
    disp('Since It does not satisify the
       superposition principle')
    disp('The given system is a Non-Linear system')
end
```

Scilab code Exa 1.17

Linearity Property

```
//clear//
//Example 1.17:Classification of system:Linearity
   Property
//Page 54
//To check whether the given discrete system is a
   Linear System (or) Non-Linear System
//Given discrete system y(t)= t*x(t)
clear;
clc;
x1 = [1,1,1,1];
x2 = [2,2,2,2];
a = 1;
b = 1;
for t = 1:length(x1)
  x3(t) = a*x1(t)+b*x2(t);
end
for t = 1:length(x1)
  y1(t) = t*x1(t);
  y2(t) = t*x2(t);
  y3(t) = t*x3(t);
end
for t = 1:length(y1)
```

```
  z(t) = a*y1(t)+b*y2(t);
end
count = 0;
for n =1:length(y1)
  if(y3(t)== z(t))
    count = count+1;
  end
end
if(count == length(y3))
   disp('Since It satisifies the superposition
      principle')
   disp('The given system is a Linear system')
   y3
   z
  else
    disp('Since It does not satisify the
       superposition principle')
    disp('The given system is a Non-Linear system')
end
```

Scilab code Exa 1.18 Linearity Property

```
//clear//
//Example 1.18:Classsification of a system:Linearity
    Property
//Page 54
//To check whether the given discrete system is a
   Linear System (or) Non-Linear System
//Given discrete system y(t)= (x(t)^2)
clear;
clc;
x1 = [1,1,1,1];
x2 = [2,2,2,2];
```

```
a = 1;
b = 1;
for t = 1:length(x1)
  x3(t) = a*x1(t)+b*x2(t);
end
for t = 1:length(x1)
  y1(t) = (x1(t)^2);
  y2(t) = (x2(t)^2);
  y3(t) = (x3(t)^2);
end
for t = 1:length(y1)
  z(t) = a*y1(t)+b*y2(t);
end
count = 0;
for n =1:length(y1)
  if(y3(t)== z(t))
    count = count+1;
  end
end
if(count == length(y3))
   disp('Since It satisifies the superposition
      principle')
   disp('The given system is a Linear system')
   y3
   z
  else
    disp('Since It does not satisify the
       superposition principle')
    disp('The given system is a Non-Linear system')
end
```

Scilab code Exa 1.18

Linearity Property

```
//clear//
//Example 1.18:Classsification of a system:Linearity
   Property
```

```
//Page 54
//To check whether the given discrete system is a
   Linear System (or) Non-Linear System
//Given discrete system y(t)= (x(t)^2)
clear;
clc;
x1 = [1,1,1,1];
x2 = [2,2,2,2];
a = 1;
b = 1;
for t = 1:length(x1)
  x3(t) = a*x1(t)+b*x2(t);
end
for t = 1:length(x1)
  y1(t) = (x1(t)^2);
  y2(t) = (x2(t)^2);
  y3(t) = (x3(t)^2);
end
for t = 1:length(y1)
  z(t) = a*y1(t)+b*y2(t);
end
count = 0;
for n =1:length(y1)
  if(y3(t)== z(t))
    count = count+1;
  end
end
if(count == length(y3))
   disp('Since It satisifies the superposition
      principle')
   disp('The given system is a Linear system')
   y3
   z
  else
    disp('Since It does not satisify the
       superposition principle')
    disp('The given system is a Non-Linear system')
end
```

Scilab code Exa 1.20 Linearity Property

```
//clear//
//Example 1.20:Classsification of a system:Linearity
    Property
//Page 55
//To check whether the given discrete system is a
   Linear System (or) Non-Linear System
//Given discrete system y[n])= 2*x[n]+3
clear;
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) = 2*x1(n)+3;
  y2(n) = 2*x2(n)+3;
  y3(n) = 2*x3(n)+3;
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('Since It satisifies the superposition
      principle')
   disp('The given system is a Linear system')
   y3
   z
  else
    disp('Since It does not satisify the
       superposition principle')
    disp('The given system is a Non-Linear system')
end
```

Scilab code Exa 1.20

Linearity Property

```
//clear//
//Example 1.20:Classsification of a system:Linearity
    Property
//Page 55
//To check whether the given discrete system is a
   Linear System (or) Non-Linear System
//Given discrete system y[n])= 2*x[n]+3
clear;
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) = 2*x1(n)+3;
  y2(n) = 2*x2(n)+3;
  y3(n) = 2*x3(n)+3;
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('Since It satisifies the superposition
      principle')
   disp('The given system is a Linear system')
   y3
   z
  else
    disp('Since It does not satisify the
       superposition principle')
    disp('The given system is a Non-Linear system')
end
```

Chapter 3

Linear Time Invariant Systems

Chapter 4

Linear Time Invariant Systems

Scilab code Exa 2.1 Linear Convolution Sum

```
//clear//
//Example 2.1:Linear Convolution Sum
//page 80
clear;
close;
clc;
h = [0,0,1,1,1,0,0];
N1 = -2:4;
x = [0,0,0.5,2,0,0,0];
N2 = -2:4;
y = convol(x,h);
for i = 1:length(y)
  if (y(i)<=0.0001)
    y(i)=0;
  end
end
N = -4:8;
figure
a=gca();
```

```
plot2d3('gnn',N1,h)
xtitle('Impulse Response','n','h[n]');
a.thickness = 2;
figure
a=gca();
plot2d3('gnn',N2,x)
xtitle('Input Response','n','x[n]');
a.thickness = 2;
figure
a=gca();
plot2d3('gnn',N,y)
xtitle('Output Response','n','y[n]');
a.thickness = 2;
```

Scilab code Exa 2.1

Linear Convolution Sum

```
//clear//
//Example 2.1:Linear Convolution Sum
//page 80
clear;
close;
clc;
h = [0,0,1,1,1,0,0];
N1 = -2:4;
x = [0,0,0.5,2,0,0,0];
N2 = -2:4;
y = convol(x,h);
for i = 1:length(y)
  if (y(i)<=0.0001)
    y(i)=0;
  end
end
N = -4:8;
figure
a=gca();
plot2d3('gnn',N1,h)
```

```
xtitle('Impulse Response','n','h[n]');
a.thickness = 2;
figure
a=gca();
plot2d3('gnn',N2,x)
xtitle('Input Response','n','x[n]');
a.thickness = 2;
figure
a=gca();
plot2d3('gnn',N,y)
xtitle('Output Response','n','y[n]');
a.thickness = 2;
```

Scilab code Exa 2.3 Scilab code Exa 2.3 Convolution of x[n] and Unit Impulse response h[n]

```
//clear//
//Example 2.3:Convolution Sum:Convolution of x[n]
    and
//Unit Impulse response h[n]
clear;
close;
clc;
Max_Limit = 10;
h = ones(1,Max_Limit);
N1 = 0:Max_Limit-1;
Alpha = 0.5;     //alpha < 1
for n = 1:Max_Limit
  x(n)= (Alpha^(n-1))*1;
end
N2 = 0:Max_Limit-1;
y = convol(x,h);
N = 0:2*Max_Limit-2;
```

```
figure
a=gca();
plot2d3('gnn',N1,h)
xtitle('Impulse Response Fig 2.5.(b)','n','h[n]');
a.thickness = 2;
figure
a=gca();
plot2d3('gnn',N2,x)
xtitle('Input Response Fig 2.5.(a)','n','x[n]');
a.thickness = 2;
figure
a=gca();
plot2d3('gnn',N(1:Max_Limit),y(1:Max_Limit),5)
xtitle('Output Response Fig 2.7','n','y[n]');
a.thickness = 2;
```

Convolution of x[n] and Unit Impulse response h[n]

```
//clear//
//Example 2.3:Convolution Sum:Convolution of x[n]
   and
//Unit Impulse response h[n]
clear;
close;
clc;
Max_Limit = 10;
h = ones(1,Max_Limit);
N1 = 0:Max_Limit-1;
Alpha = 0.5;     //alpha < 1
for n = 1:Max_Limit
  x(n)= (Alpha^(n-1))*1;
end
N2 = 0:Max_Limit-1;
y = convol(x,h);
N = 0:2*Max_Limit-2;
figure
a=gca();
plot2d3('gnn',N1,h)
```

```
xtitle('Impulse Response Fig 2.5.(b)','n','h[n]');
a.thickness = 2;
figure
a=gca();
plot2d3('gnn',N2,x)
xtitle('Input Response Fig 2.5.(a)','n','x[n]');
a.thickness = 2;
figure
a=gca();
plot2d3('gnn',N(1:Max_Limit),y(1:Max_Limit),5)
xtitle('Output Response Fig 2.7','n','y[n]');
a.thickness = 2;
```

Scilab code Exa 2.4 Convolution Sum of finite duration sequences

```
//clear//
//Example 2.4:Convolution Sum of finite duration
   sequences
clear;
close;
clc;
x = ones(1,5);
N1 =0:length(x)-1;
Alpha = 1.4;     //alpha > 1
for n = 1:7
  h(n)= (Alpha^(n-1))*1;
end
N2 =0:length(h)-1;
y = convol(x,h);
N = 0:length(x)+length(h)-2;
figure
a=gca();
plot2d3('gnn',N2,h)
```

```
xtitle('Impulse Response','n','h[n]');
a.thickness = 2;
figure
a=gca();
plot2d3('gnn',N1,x)
xtitle('Input Response','n','x[n]');
a.thickness = 2;
figure
a=gca();
plot2d3('gnn',N,y)
xtitle('Output Response','n','y[n]');
a.thickness = 2;
```

Scilab code Exa 2.4

Convolution Sum of finite duration sequences

```
//clear//
//Example 2.4:Convolution Sum of finite duration
   sequences
clear;
close;
clc;
x = ones(1,5);
N1 =0:length(x)-1;
Alpha = 1.4;     //alpha > 1
for n = 1:7
  h(n)= (Alpha^(n-1))*1;
end
N2 =0:length(h)-1;
y = convol(x,h);
N = 0:length(x)+length(h)-2;
figure
a=gca();
plot2d3('gnn',N2,h)
xtitle('Impulse Response','n','h[n]');
a.thickness = 2;
figure
```

```
a=gca();
plot2d3('gnn',N1,x)
xtitle('Input Response','n','x[n]');
a.thickness = 2;
figure
a=gca();
plot2d3('gnn',N,y)
xtitle('Output Response','n','y[n]');
a.thickness = 2;
```

Scilab code Exa 2.5 Convolution Sum of input sequence

```
//clear//
//Example 2.5:Convolution Sum of input sequence x[n
    ]=(2^n).u[-n]
//and h[n] = u[n]
clear;
close;
clc;
Max_Limit = 10;
h = ones(1,Max_Limit);
N2 =0:length(h)-1;
for n = 1:Max_Limit
  x1(n)= (2^(-(n-1)))*1;
end
x = x1($:-1:1);
N1 = -length(x)+1:0;
y = convol(x,h);
N = -length(x)+1:length(h)-1;
figure
a=gca();
plot2d3('gnn',N2,h)
xtitle('Impulse Response','n','h[n]');
```

```
a.thickness = 2;
figure
a=gca();
a.y_location = "origin";
plot2d3('gnn',N1,x)
xtitle('Input Response Fig 2.11(a)','n','x[n]');
a.thickness = 2;
figure
a=gca();
a.y_location ="origin";
plot2d3('gnn',N,y)
xtitle('Output Response Fig 2.11(b)','n','y[n]');
a.thickness = 2;
```

Scilab code Exa 2.5

Convolution Sum of input sequence

```
//clear//
//Example 2.5:Convolution Sum of input sequence x[n
   ]=(2^n).u[-n]
//and h[n] = u[n]
clear;
close;
clc;
Max_Limit = 10;
h = ones(1,Max_Limit);
N2 =0:length(h)-1;
for n = 1:Max_Limit
  x1(n)= (2^(-(n-1)))*1;
end
x = x1($:-1:1);
N1 = -length(x)+1:0;
y = convol(x,h);
N = -length(x)+1:length(h)-1;
figure
a=gca();
plot2d3('gnn',N2,h)
```

```
xtitle('Impulse Response','n','h[n]');
a.thickness = 2;
figure
a=gca();
a.y_location = "origin";
plot2d3('gnn',N1,x)
xtitle('Input Response Fig 2.11(a)','n','x[n]');
a.thickness = 2;
figure
a=gca();
a.y_location ="origin";
plot2d3('gnn',N,y)
xtitle('Output Response Fig 2.11(b)','n','y[n]');
a.thickness = 2;
```

Scilab code Exa 2.6 Convolution Integral of input

```
//clear//
//Example 2.6:Convolution Integral of input x(t) = (
    e^-at).u(t)
//and h(t) =u(t)
clear;
close;
clc;
Max_Limit = 10;
h = ones(1,Max_Limit);
N2 =0:length(h)-1;
a = 0.5;  //constant a>0
for t = 1:Max_Limit
  x(t)= exp(-a*(t-1));
end
N1 =0:length(x)-1;
y = convol(x,h)-1;
```

```
N = 0:length(x)+length(h)-2;
figure
a=gca();
plot2d(N2,h)
xtitle('Impulse Response','t','h(t)');
a.thickness = 2;
figure
a=gca();
plot2d(N1,x)
xtitle('Input Response','t','x(t)');
a.thickness = 2;
figure
a=gca();
plot2d(N(1:Max_Limit),y(1:Max_Limit))
xtitle('Output Response','t','y(t)');
a.thickness = 2;
```

Scilab code Exa 2.6

Convolution Integral of input

```
//clear//
//Example 2.6:Convolution Integral of input x(t) = (
    e^-at).u(t)
//and h(t) =u(t)
clear;
close;
clc;
Max_Limit = 10;
h = ones(1,Max_Limit);
N2 =0:length(h)-1;
a = 0.5;  //constant a>0
for t = 1:Max_Limit
  x(t)= exp(-a*(t-1));
end
N1 =0:length(x)-1;
y = convol(x,h)-1;
N = 0:length(x)+length(h)-2;
```

```
figure
a=gca();
plot2d(N2,h)
xtitle('Impulse Response','t','h(t)');
a.thickness = 2;
figure
a=gca();
plot2d(N1,x)
xtitle('Input Response','t','x(t)');
a.thickness = 2;
figure
a=gca();
plot2d(N(1:Max_Limit),y(1:Max_Limit))
xtitle('Output Response','t','y(t)');
a.thickness = 2;
```

Scilab code Exa 2.7 Convolution Integral of fintie duration signals

Scilab code Exa 2.7

```
//clear//
//Example 2.7:Convolution Integral of fintie
   duration signals
//page99
clear;
close;
clc;
T = 10;
x = ones(1,T);  //Input Response
for t = 1:2*T
  h(t) = t-1;    //Impulse Response
end
N1 = 0:length(x)-1;
N2 = 0:length(h)-1;
```

```
y = convol(x,h);
N = 0:length(x)+length(h)-2;
figure
a=gca();
a.x_location="origin";
plot2d(N2,h)
xtitle('Impulse Response','t','h(t)');
a.thickness = 2;
figure
a=gca();
plot2d(N1,x)
xtitle('Input Response','t','x(t)');
a.thickness = 2;
figure
a=gca();
plot2d(N,y)
xtitle('Output Response','t','y(t)');
a.thickness = 2;
```

Convolution Integral of fintie duration signals

```
//clear//
//Example 2.7:Convolution Integral of fintie
   duration signals
//page99
clear;
close;
clc;
T = 10;
x = ones(1,T);  //Input Response
for t = 1:2*T
  h(t) = t-1;     //Impulse Response
end
N1 = 0:length(x)-1;
N2 = 0:length(h)-1;
y = convol(x,h);
N = 0:length(x)+length(h)-2;
figure
```

```
a=gca();
a.x_location="origin";
plot2d(N2,h)
xtitle('Impulse Response','t','h(t)');
a.thickness = 2;
figure
a=gca();
plot2d(N1,x)
xtitle('Input Response','t','x(t)');
a.thickness = 2;
figure
a=gca();
plot2d(N,y)
xtitle('Output Response','t','y(t)');
a.thickness = 2;
```

Scilab code Exa 2.8 Convolution Integral of input

```
//clear//
//Example 2.8:Convolution Integral of input x(t)=(e
    ^2t).u(-t) and
//h(t) = u(t-3)
clear;
close;
clc;
Max_Limit = 10;
h =[0,0,0,ones(1,Max_Limit-3)]; //h(n-3)
a = 2;
t = -9:0;
x= exp(a*t);
//x = x1($:-1:1)
N2 = 0:length(h)-1;
N1 = -length(x)+1:0;
```

```
t1 = -6:3;
y1 = (1/a)*exp(a*(t1-3));
y2 = (1/a)*ones(1,Max_Limit);
y = [y1 y2]
N = -length(h)+1:length(x)-1;
figure
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d(-Max_Limit+1:0,h($:-1:1))
xtitle('Impulse Response','t','h(t-T)');
a.thickness = 2;
figure
a=gca();
a.y_location = "origin";
plot2d(t,x)
xtitle('Input Response','t','x(t)');
a.thickness = 2;
figure
a=gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-10,0;13,1];
plot2d(-Max_Limit+4:Max_Limit+3,y)
xtitle('Output Response','t','y(t)');
a.thickness = 2;
```

Scilab code Exa 2.8

Convolution Integral of input

```
//clear//
//Example 2.8:Convolution Integral of input x(t)=(e
    ^2t).u(-t) and
//h(t) = u(t-3)
clear;
close;
clc;
```

```
Max_Limit = 10;
h =[0,0,0,ones(1,Max_Limit-3)]; //h(n-3)
a = 2;
t = -9:0;
x= exp(a*t);
//x = x1($:-1:1)
N2 = 0:length(h)-1;
N1 = -length(x)+1:0;
t1 = -6:3;
y1 = (1/a)*exp(a*(t1-3));
y2 = (1/a)*ones(1,Max_Limit);
y = [y1 y2]
N = -length(h)+1:length(x)-1;
figure
a=gca();
a.x_location="origin";
a.y_location="origin";
plot2d(-Max_Limit+1:0,h($:-1:1))
xtitle('Impulse Response','t','h(t-T)');
a.thickness = 2;
figure
a=gca();
a.y_location = "origin";
plot2d(t,x)
xtitle('Input Response','t','x(t)');
a.thickness = 2;
figure
a=gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-10,0;13,1];
plot2d(-Max_Limit+4:Max_Limit+3,y)
xtitle('Output Response','t','y(t)');
a.thickness = 2;
```

Chapter 5

Fourier Series Repreentation of Periodic Signals

Chapter 6

Fourier Series Repreentation of Periodic Signals

Scilab code Exa 3.2 CTFS of a periodic signal x(t)

```
//clear//
//Example 3.2:CTFS of a periodic signal x(t)
//Expression of continuous time signal
//using continuous time fourier series
clear;
close;
clc;
t = -3:0.01:3;
//t1 = -%pi*4:(%pi*4)/100:%pi*4;
//t2 =-%pi*6:(%pi*6)/100:%pi*6;
xot = ones(1,length(t));
x1t = (1/2)*cos(%pi*2*t);
xot_x1t = xot+x1t;
x2t = cos(%pi*4*t);
xot_x1t_x2t = xot+x1t+x2t;
x3t = (2/3)*cos(%pi*6*t);
xt = xot+x1t+x2t+x3t;
```

```
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,0;2 4];
plot(t,xot)
ylabel('t')
title('xot =1')
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,-3;2 4];
plot(t,x1t)
ylabel('t')
title('x1(t) =1/2*cos(2*pi*t)')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,0;2 4];
plot(t,xot_x1t)
ylabel('t')
title('xo(t)+x1(t)')
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,-2;4 2];
plot(t,x2t)
ylabel('t')
title('x2(t) =cos(4*pi*t)')
subplot(2,1,2)
```

```
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,0;4 4];
plot(t,xot_x1t_x2t)
ylabel('t')
title('xo(t)+x1(t)+x2(t)')
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,-3;4 3];
plot(t,x3t)
ylabel('t')
title('x1(t) =2/3*cos(6*pi*t)')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,-3;4 3];
plot(t,xt)
ylabel('t')
title('x(t)=xo(t)+x1(t)+x2(t)+x3(t)')
```

Scilab code Exa 3.2

CTFS of a periodic signal x(t)

```
//clear//
//Example 3.2:CTFS of a periodic signal x(t)
//Expression of continuous time signal
//using continuous time fourier series
clear;
close;
clc;
t = -3:0.01:3;
```

```
//t1 = −%pi∗4:(%pi∗4)/100:%pi∗4;
//t2 =−%pi∗6:(%pi∗6)/100:%pi∗6;
xot = ones(1,length(t));
x1t = (1/2)*cos(%pi*2*t);
xot_x1t = xot+x1t;
x2t = cos(%pi*4*t);
xot_x1t_x2t = xot+x1t+x2t;
x3t = (2/3)*cos(%pi*6*t);
xt = xot+x1t+x2t+x3t;
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,0;2 4];
plot(t,xot)
ylabel('t')
title('xot =1')
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,-3;2 4];
plot(t,x1t)
ylabel('t')
title('x1(t) =1/2∗cos(2∗pi∗t)')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,0;2 4];
plot(t,xot_x1t)
ylabel('t')
title('xo(t)+x1(t)')
//
figure
```

```
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,-2;4 2];
plot(t,x2t)
ylabel('t')
title('x2(t) =cos(4*pi*t)')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,0;4 4];
plot(t,xot_x1t_x2t)
ylabel('t')
title('xo(t)+x1(t)+x2(t)')
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,-3;4 3];
plot(t,x3t)
ylabel('t')
title('x1(t) =2/3*cos(6*pi*t)')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-4,-3;4 3];
plot(t,xt)
ylabel('t')
title('x(t)=xo(t)+x1(t)+x2(t)+x3(t)')
```

Scilab code Exa 3.3 Continuous Time Fourier Series Coefficients

```
//clear//
//Example3.3:Continuous Time Fourier Series
   Coefficients of
//a periodic signal x(t) = sin(Wot)
clear;
close;
clc;
t = 0:0.01:1;
T = 1;
Wo = 2*%pi/T;
xt = sin(Wo*t);
for k =0:5
  C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
  a(k+1) = xt*C(k+1,:)'/length(t);
  if(abs(a(k+1))<=0.01)
    a(k+1)=0;
  end
end
a =a';
ak = [-a,a(2:$)];
```

Scilab code Exa 3.3

Continuous Time Fourier Series Coefficients

```
//clear//
//Example3.3:Continuous Time Fourier Series
   Coefficients of
//a periodic signal x(t) = sin(Wot)
clear;
close;
clc;
t = 0:0.01:1;
T = 1;
```

```
Wo = 2*%pi/T;
xt = sin(Wo*t);
for k =0:5
  C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
  a(k+1) = xt*C(k+1,:)'/length(t);
  if(abs(a(k+1))<=0.01)
    a(k+1)=0;
  end
end
a =a';
ak = [-a,a(2:$)];
```

Scilab code Exa 3.4 CTFS coefficients of a periodic signal

```
//clear//
//Example3.4:CTFS coefficients of a periodic signal
//x(t) = 1+sin(Wot)+2cos(Wot)+cos(2Wot+%pi/4)
clear;
close;
clc;
t = 0:0.01:1;
T = 1;
Wo = 2*%pi/T;
xt =ones(1,length(t))+sin(Wo*t)+2*cos(Wo*t)+cos(2*Wo
   *t+%pi/4);
for k =0:5
  C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
  a(k+1) = xt*C(k+1,:)'/length(t);
  if(abs(a(k+1))<=0.1)
    a(k+1)=0;
  end
end
a =a';
```

```
a_conj =conj(a);
ak = [a_conj($:-1:1),a(2:$)];
Mag_ak = abs(ak);
for i = 1:length(a)
  Phase_ak(i) = atan(imag(ak(i))/(real(ak(i))
     +0.0001));
end
Phase_ak = Phase_ak'
Phase_ak = [Phase_ak(1:$) -Phase_ak($-1:-1:1)];
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',[-k:k],Mag_ak,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('abs(ak)')
xlabel('

   k')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',[-k:k],Phase_ak,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('<(ak)')
xlabel('

   k')
```

Scilab code Exa 3.4

CTFS coefficients of a periodic signal

```
//clear//
```

```
//Example3.4:CTFS coefficients of a periodic signal
//x(t) = 1+sin(Wot)+2cos(Wot)+cos(2Wot+%pi/4)
clear;
close;
clc;
t = 0:0.01:1;
T = 1;
Wo = 2*%pi/T;
xt =ones(1,length(t))+sin(Wo*t)+2*cos(Wo*t)+cos(2*Wo
   *t+%pi/4);
for k =0:5
  C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
  a(k+1) = xt*C(k+1,:)'/length(t);
  if(abs(a(k+1))<=0.1)
    a(k+1)=0;
  end
end
a =a';
a_conj =conj(a);
ak = [a_conj($:-1:1),a(2:$)];
Mag_ak = abs(ak);
for i = 1:length(a)
  Phase_ak(i) = atan(imag(ak(i))/(real(ak(i))
     +0.0001));
end
Phase_ak = Phase_ak'
Phase_ak = [Phase_ak(1:$) -Phase_ak($-1:-1:1)];
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',[-k:k],Mag_ak,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('abs(ak)')
xlabel('
   k')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',[-k:k],Phase_ak,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('<(ak)')
xlabel('

   k')
```

Scilab code Exa 3.5 CTFS coefficients of a periodic signal

```
//clear//
//Example3.5:CTFS coefficients of a periodic signal
//x(t) = 1, |t|<T1, and 0, T1<|t|<T/2
clear;
close;
clc;
T =4;
T1 = T/4;
t = -T1:T1/100:T1;
Wo = 2*%pi/T;
xt =ones(1,length(t));
//
for k =0:5
  C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
  a(k+1) = xt*C(k+1,:)'/length(t);
  if(abs(a(k+1))<=0.1)
    a(k+1)=0;
  end
```

```
end
a =a';
a_conj = real(a(:))-sqrt(-1)*imag(a(:));
ak = [a_conj($:-1:1)',a(2:$)];
k = 0:5;
k = [-k($:-1:1),k(2:$)];
Spectrum_ak = (1/2)*real(ak);
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-2,0;2,2];
plot2d(t,xt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x(t)')
xlabel('

    t')
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,Spectrum_ak,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('abs(ak)')
xlabel('

    k')
```

Scilab code Exa 3.5

CTFS coefficients of a periodic signal

```
//clear//
```

```
//Example3.5:CTFS coefficients of a periodic signal
//x(t) = 1, |t|<T1, and 0, T1<|t|<T/2
clear;
close;
clc;
T =4;
T1 = T/4;
t = -T1:T1/100:T1;
Wo = 2*%pi/T;
xt =ones(1,length(t));
//
for k =0:5
  C(k+1,:) = exp(-sqrt(-1)*Wo*t.*k);
  a(k+1) = xt*C(k+1,:)'/length(t);
  if(abs(a(k+1))<=0.1)
    a(k+1)=0;
  end
end
a =a';
a_conj = real(a(:))-sqrt(-1)*imag(a(:));
ak = [a_conj($:-1:1)',a(2:$)];
k = 0:5;
k = [-k($:-1:1),k(2:$)];
Spectrum_ak = (1/2)*real(ak);
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-2,0;2,2];
plot2d(t,xt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x(t)')
xlabel('

   t')
//
```

```
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,Spectrum_ak,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('abs(ak)')
xlabel('

    k')
```

Scilab code Exa 3.6 Time Shift Property of CTFS

```
//clear//
//Example3.6: Time Shift Property of CTFS
clear;
close;
clc;
T =4;
T1 = T/2;
t = 0:T1/100:T1;
Wo = 2*%pi/T;
gt =(1/2)*ones(1,length(t));
a(1)=0; //k=0, ak =0
d(1)=0;
for k =1:5
  a(k+1) = (sin(%pi*k/2)/(k*%pi));
  if(abs(a(k+1))<=0.01)
    a(k+1)=0;
  end
   d(k+1) = a(k+1)*exp(-sqrt(-1)*k*%pi/2);
end
```

```
k = 0:5
disp('Fourier Series Coefficients of Square Wave')
a
disp('Fourier Series Coefficients of g(t)=x(t-1)-0.5
    ')
d
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-1,-2;1,4];
plot2d([-t($:-1:1),t(1:$)],[-gt,gt],5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('g(t)')
xlabel('

    t')
```

Scilab code Exa 3.6

Time Shift Property of CTFS

```
//clear//
//Example3.6: Time Shift Property of CTFS
clear;
close;
clc;
T =4;
T1 = T/2;
t = 0:T1/100:T1;
Wo = 2*%pi/T;
gt =(1/2)*ones(1,length(t));
a(1)=0; //k=0, ak =0
d(1)=0;
for k =1:5
  a(k+1) = (sin(%pi*k/2)/(k*%pi));
  if(abs(a(k+1))<=0.01)
    a(k+1)=0;
  end
   d(k+1) = a(k+1)*exp(-sqrt(-1)*k*%pi/2);
end
k = 0:5
disp('Fourier Series Coefficients of Square Wave')
a
disp('Fourier Series Coefficients of g(t)=x(t-1)-0.5
    ')
d
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-1,-2;1,4];
plot2d([-t($:-1:1),t(1:$)],[-gt,gt],5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('g(t)')
xlabel('

    t')
```

Scilab code Exa 3.7 Derivative Property of CTFS

```
//clear//
//Example3.7:Derivative Property of CTFS
clear;
clc;
close;
T =4;
```

```
T1 = T/2;
t = 0:T1/100:T1;
xt = [t($:-1:1) t]/T1;
gt =(1/2)*ones(1,length(t));
e(1) = 1/2; //k =0, e0 = 1/2
for k =1:5
  a(k+1) = (sin(%pi*k/2)/(k*%pi));
  if(abs(a(k+1))<=0.01)
    a(k+1)=0;
  end
  d(k+1) = a(k+1)*exp(-sqrt(-1)*k*%pi/2);
  e(k+1) = 2*d(k+1)/(sqrt(-1)*k*%pi);
end
k = 0:5
disp('Fourier Series Coefficients of Square Wave')
a
disp('Fourier Series Coefficients of g(t)=x(t-1)-0.5
   ')
d
disp('Fourier Series Coefficients of Triangular Wave
   ')
e
//Plotting the time shifted square waveform
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-1,-2;1,2];
plot2d([-t($:-1:1),t(1:$)],[-gt,gt],5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('g(t)')
xlabel('

   t')
//Plotting the Triangular waveform
figure
a = gca();
```

```
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-1,0;1,2];
plot2d([-t($:-1:1),t(1:$)],xt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x(t)')
xlabel('t')
```

Scilab code Exa 3.7

Derivative Property of CTFS

```
//clear//
//Example3.7:Derivative Property of CTFS
clear;
clc;
close;
T =4;
T1 = T/2;
t = 0:T1/100:T1;
xt = [t($:-1:1) t]/T1;
gt =(1/2)*ones(1,length(t));
e(1) = 1/2; //k =0, e0 = 1/2
for k =1:5
  a(k+1) = (sin(%pi*k/2)/(k*%pi));
  if(abs(a(k+1))<=0.01)
    a(k+1)=0;
  end
  d(k+1) = a(k+1)*exp(-sqrt(-1)*k*%pi/2);
  e(k+1) = 2*d(k+1)/(sqrt(-1)*k*%pi);
end
k = 0:5
disp('Fourier Series Coefficients of Square Wave')
a
disp('Fourier Series Coefficients of g(t)=x(t-1)-0.5
    ')
d
```

```
disp('Fourier Series Coefficients of Triangular Wave
   ')
e
//Plotting the time shifted square waveform
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-1,-2;1,2];
plot2d([-t($:-1:1),t(1:$)],[-gt,gt],5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('g(t)')
xlabel('

   t')
//Plotting the Triangular waveform
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-1,0;1,2];
plot2d([-t($:-1:1),t(1:$)],xt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x(t)')
xlabel('t')
```

Scilab code Exa 3.8 Fourier Series Representation of Periodic Impulse Train

```
//clear//
//Example3.8:Fourier Series Representation of
```

```
    Periodic Impulse Train
clear;
clc;
close;
T =4;
T1 = T/4;
t = [-T,0,T];
xt = [1,1,1]; //Generation of Periodic train of
    Impulses
t1 = -T1:T1/100:T1;
gt = ones(1,length(t1));//Generation of periodic
    square wave
t2 = [-T1,0,T1];
qt = [1,0,-1];//Derivative of periodic square wave
Wo = 2*%pi/T;
ak = 1/T;
b(1) = 0;
c(1) = 2*T1/T;
for k =1:5
  b(k+1) = ak*(exp(sqrt(-1)*k*Wo*T1)-exp(-sqrt(-1)*k
     *Wo*T1));
  if(abs(b(k+1))<=0.1)
    b(k+1) =0;
  end
  c(k+1) = b(k+1)/(sqrt(-1)*k*Wo);
  if(abs(c(k+1))<=0.1)
    c(k+1) =0;
  end
end
k = 0:5
disp('Fourier Series Coefficients of periodic Square
     Wave')
disp(b)
disp('Fourier Series Coefficients of derivative of
    periodic square wave')
disp(c)
//Plotting the periodic train of impulses
figure
```

```
subplot(3,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-6,0;6,2];
plot2d3('gnn',t,xt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x(t)')
//Plotting the periodic square waveform
subplot(3,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-6,0;6,2];
plot2d(t1,gt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d(T+t1,gt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d(-T+t1,gt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('g(t)')
//Plotting the periodic square waveform
subplot(3,1,3)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-6,-2;6,2];
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',t2,qt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',T+t2,qt,5)
poly1 = a.children(1).children(1);
```

```
poly1.thickness = 3;
plot2d3('gnn',-T+t2,qt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('q(t)')
```

Scilab code Exa 3.8

Fourier Series Representation of Periodic Impulse Train

```
//clear//
//Example3.8:Fourier Series Representation of
   Periodic Impulse Train
clear;
clc;
close;
T =4;
T1 = T/4;
t = [-T,0,T];
xt = [1,1,1]; //Generation of Periodic train of
   Impulses
t1 = -T1:T1/100:T1;
gt = ones(1,length(t1));//Generation of periodic
   square wave
t2 = [-T1,0,T1];
qt = [1,0,-1];//Derivative of periodic square wave
Wo = 2*%pi/T;
ak = 1/T;
b(1) = 0;
c(1) = 2*T1/T;
for k =1:5
  b(k+1) = ak*(exp(sqrt(-1)*k*Wo*T1)-exp(-sqrt(-1)*k
     *Wo*T1));
  if(abs(b(k+1))<=0.1)
    b(k+1) =0;
  end
  c(k+1) = b(k+1)/(sqrt(-1)*k*Wo);
  if(abs(c(k+1))<=0.1)
```

```
    c(k+1) =0;
  end
end
k = 0:5
disp('Fourier Series Coefficients of periodic Square
    Wave')
disp(b)
disp('Fourier Series Coefficients of derivative of
   periodic square wave')
disp(c)
//Plotting the periodic train of impulses
figure
subplot(3,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-6,0;6,2];
plot2d3('gnn',t,xt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x(t)')
//Plotting the periodic square waveform
subplot(3,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-6,0;6,2];
plot2d(t1,gt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d(T+t1,gt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d(-T+t1,gt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('g(t)')
//Plotting the periodic square waveform
```

```
subplot(3,1,3)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-6,-2;6,2];
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',t2,qt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',T+t2,qt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',-T+t2,qt,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('q(t)')
```

Scilab code Exa 3.10 **Scilab code Exa 3.10** DTFS of x[n]

```
//clear//
//Example3.10:DTFS of x[n] =sin(Won)
clear;
close;
clc;
n = 0:0.01:5;
N = 5;
Wo = 2*%pi/N;
xn = sin(Wo*n);
for k =0:N-2
  C(k+1,:) = exp(-sqrt(-1)*Wo*n.*k);
  a(k+1) = xn*C(k+1,:)'/length(n);
  if(abs(a(k+1))<=0.01)
```

```
    a(k+1)=0;
  end
end
a =a'
a_conj = conj(a);
ak = [a_conj($:-1:1),a(2:$)]
k = -(N-2):(N-2);
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-8,-1;8,1];
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',k,-imag(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',N+k,-imag(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',-(N+k),-imag(ak($:-1:1)),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('ak')
```

DTFS of x[n]

```
//clear//
//Example3.10:DTFS of x[n] =sin(Won)
clear;
close;
clc;
n = 0:0.01:5;
N = 5;
Wo = 2*%pi/N;
xn = sin(Wo*n);
for k =0:N-2
```

```
  C(k+1,:) = exp(-sqrt(-1)*Wo*n.*k);
  a(k+1) = xn*C(k+1,:)'/length(n);
  if(abs(a(k+1))<=0.01)
    a(k+1)=0;
  end
end
a =a'
a_conj = conj(a);
ak = [a_conj($:-1:1),a(2:$)]
k = -(N-2):(N-2);
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
a.data_bounds=[-8,-1;8,1];
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',k,-imag(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',N+k,-imag(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',-(N+k),-imag(ak($:-1:1)),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('ak')
```

Scilab code Exa 3.11 Scilab code Exa 3.11 DTFS of x[n]

```
//clear//
//Example3.11:DTFS of
```

```
//x[n] = 1+sin(2*%pi/N)n+3cos(2*%pi/N)n+cos[(4*%pi/N
   )n+%pi/2]
clear;
close;
clc;
N = 10;
n = 0:0.01:N;
Wo = 2*%pi/N;
xn =ones(1,length(n))+sin(Wo*n)+3*cos(Wo*n)+cos(2*Wo
   *n+%pi/2);
for k =0:N-2
  C(k+1,:) = exp(-sqrt(-1)*Wo*n.*k);
  a(k+1) = xn*C(k+1,:)'/length(n);
  if(abs(a(k+1))<=0.1)
    a(k+1)=0;
  end
end
a =a';
a_conj =conj(a);
ak = [a_conj($:-1:1),a(2:$)];
Mag_ak = abs(ak);
for i = 1:length(a)
  Phase_ak(i) = atan(imag(ak(i))/(real(ak(i))
     +0.0001));
end
Phase_ak = Phase_ak'
Phase_ak = [Phase_ak(1:$-1) -Phase_ak($:-1:1)];
k = -(N-2):(N-2);
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,real(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('Real part of(ak)')
```

```
xlabel('

   k')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,imag(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('imaginary part of(ak)')
xlabel('

   k')
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,Mag_ak,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('abs(ak)')
xlabel('

   k')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,Phase_ak,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('<(ak)')
xlabel('

   k')
```

DTFS of x[n]

```
//clear//
//Example3.11:DTFS of
//x[n] = 1+sin(2*%pi/N)n+3cos(2*%pi/N)n+cos[(4*%pi/N
   )n+%pi/2]
clear;
close;
clc;
N = 10;
n = 0:0.01:N;
Wo = 2*%pi/N;
xn =ones(1,length(n))+sin(Wo*n)+3*cos(Wo*n)+cos(2*Wo
   *n+%pi/2);
for k =0:N-2
  C(k+1,:) = exp(-sqrt(-1)*Wo*n.*k);
  a(k+1) = xn*C(k+1,:)'/length(n);
  if(abs(a(k+1))<=0.1)
    a(k+1)=0;
  end
end
a =a';
a_conj =conj(a);
ak = [a_conj($:-1:1),a(2:$)];
Mag_ak = abs(ak);
for i = 1:length(a)
  Phase_ak(i) = atan(imag(ak(i))/(real(ak(i))
     +0.0001));
end
Phase_ak = Phase_ak'
Phase_ak = [Phase_ak(1:$-1) -Phase_ak($:-1:1)];
k = -(N-2):(N-2);
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
```

```
a.x_location = "origin";
plot2d3('gnn',k,real(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('Real part of(ak)')
xlabel('

   k')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,imag(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('imaginary part of(ak)')
xlabel('

   k')
//
figure
subplot(2,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,Mag_ak,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('abs(ak)')
xlabel('

   k')
subplot(2,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,Phase_ak,5)
poly1 = a.children(1).children(1);
```

```
poly1.thickness = 3;
title('<(ak)')
xlabel('

   k')
```

Scilab code Exa 3.12 DTFS coefficients of periodic square wave

```
//clear//
//Example3.12:DTFS coefficients of periodic square
   wave
clear;
close;
clc;
N = 10;
N1 = 2;
Wo = 2*%pi/N;
xn = ones(1,length(N));
n = -(2*N1+1):(2*N1+1);
a(1) = (2*N1+1)/N;
for k =1:2*N1
  a(k+1) = sin((2*%pi*k*(N1+0.5))/N)/sin(%pi*k/N);
  a(k+1) = a(k+1)/N;
  if(abs(a(k+1))<=0.1)
    a(k+1) =0;
  end
end
a =a';
a_conj =conj(a);
ak = [a_conj($:-1:1),a(2:$)];
k = -2*N1:2*N1;
//
figure
```

```
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,real(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('Real part of(ak)')
xlabel('

    k')
```

Scilab code Exa 3.12

DTFS coefficients of periodic square wave

```
//clear//
//Example3.12:DTFS coefficients of periodic square
    wave
clear;
close;
clc;
N = 10;
N1 = 2;
Wo = 2*%pi/N;
xn = ones(1,length(N));
n = -(2*N1+1):(2*N1+1);
a(1) = (2*N1+1)/N;
for k =1:2*N1
  a(k+1) = sin((2*%pi*k*(N1+0.5))/N)/sin(%pi*k/N);
  a(k+1) = a(k+1)/N;
  if(abs(a(k+1))<=0.1)
    a(k+1) =0;
  end
end
a =a';
a_conj =conj(a);
ak = [a_conj($:-1:1),a(2:$)];
k = -2*N1:2*N1;
```

```
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',k,real(ak),5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('Real part of(ak)')
xlabel('

    k')
```

Scilab code Exa 3.13 Periodic sequence

```
//clear//
//Example3.13:DTFS
//Expression of periodic sequence using
//the summation two different sequence
clear;
close;
clc;
N = 5;
n = 0:N-1;
x1 = [1,1,0,0,1];
x1 = [x1($:-1:1) x1(2:$)]; // Square Wave x1[n]
x2 = [1,1,1,1,1];
x2 = [x2($:-1:1) x2(2:$)];//DC sequence of x2[n]
x = x1+x2;//sum of x1[n] & x2[n]
//Zeroth DTFS coefficient of dc sequence
c(1) = 1;
//Zeroth DTFS coefficient of square waveform
b(1) = 3/5;
```

```
//Zeroth DTFS coefficient of sum of x1[n] & x2[n]
a(1) = b(1)+c(1);
//
Wo = 2*%pi/N;
for k =1:N-1
  a(k+1) = sin((3*%pi*k)/N)/sin(%pi*k/N);
  a(k+1) = a(k+1)/N;
  if(abs(a(k+1))<=0.1)
    a(k+1) =0;
  end
end
a =a';
a_conj =conj(a);
ak = [a_conj($:-1:1),a(2:$)];
k = -(N-1):(N-1);
n = -(N-1):(N-1);
//
figure
subplot(3,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',n,x,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x[n]')
xlabel('

   n')
subplot(3,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',n,x1,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x1[n]')
xlabel('
    n')
subplot(3,1,3)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',n,x2,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x2[n]')
xlabel('

    n')
```

Scilab code Exa 3.13

Periodic sequence

```
//clear//
//Example3.13:DTFS
//Expression of periodic sequence using
//the summation two different sequence
clear;
close;
clc;
N = 5;
n = 0:N-1;
x1 = [1,1,0,0,1];
x1 = [x1($:-1:1) x1(2:$)]; // Square Wave x1[n]
x2 = [1,1,1,1,1];
x2 = [x2($:-1:1) x2(2:$)];//DC sequence of x2[n]
x = x1+x2;//sum of x1[n] & x2[n]
//Zeroth DTFS coefficient of dc sequence
c(1) = 1;
//Zeroth DTFS coefficient of square waveform
b(1) = 3/5;
//Zeroth DTFS coefficient of sum of x1[n] & x2[n]
a(1) = b(1)+c(1);
```

```
//
Wo = 2*%pi/N;
for k =1:N-1
  a(k+1) = sin((3*%pi*k)/N)/sin(%pi*k/N);
  a(k+1) = a(k+1)/N;
  if(abs(a(k+1))<=0.1)
    a(k+1) =0;
  end
end
a =a';
a_conj =conj(a);
ak = [a_conj($:-1:1),a(2:$)];
k = -(N-1):(N-1);
n = -(N-1):(N-1);
//
figure
subplot(3,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',n,x,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x[n]')
xlabel('

   n')
subplot(3,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',n,x1,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x1[n]')
xlabel('

   n')
```

```
subplot(3,1,3)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',n,x2,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x2[n]')
xlabel('

    n')
```

Scilab code Exa 3.14 Parseval's relation of DTFS

```
//clear//
//Example3.14:DTFS
//Finding x[n] using parseval's relation of DTFS
clear;
close;
clc;
N = 6;
n = 0:N-1;
a(1) = 1/3;
a(2)=0;
a(4)=0;
a(5)=0;
a1 = (1/6)*((-1)^n);
x =0;
for k = 0:N-2
  if(k==2)
    x = x+a1;
  else
    x =  x+a(k+1);
  end
end
x = [x($:-1:1),x(2:$)];
n = -(N-1):(N-1);
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',n,x,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x[n]')
xlabel('

    n')
```

Scilab code Exa 3.14

Parseval's relation of DTFS

```
//clear//
//Example3.14:DTFS
//Finding x[n] using parseval's relation of DTFS
clear;
close;
clc;
N = 6;
n = 0:N-1;
a(1) = 1/3;
a(2)=0;
a(4)=0;
a(5)=0;
a1 = (1/6)*((-1)^n);
x =0;
for k = 0:N-2
  if(k==2)
    x = x+a1;
  else
    x =  x+a(k+1);
  end
end
x = [x($:-1:1),x(2:$)];
n = -(N-1):(N-1);
//
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',n,x,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('x[n]')
xlabel('

   n')
```

Scilab code Exa 3.15 DTFS:Periodic Convolution Property

```
//clear//
//Example3.15:DTFS:Periodic  Convolution  Property
clear;
clc;
close;
x = [1,1,0,0,0,0,1];
X = fft(x);
W = X.*X;
w = ifft(W);
w = abs(w);
for i =1:length(x)
  if (abs(w(i))<=0.1)
```

```
    w(i) = 0;
  end
end
w = [w($:-1:1) w(2:$)];
N = length(x);
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',[-(N-1):0,1:N-1],w,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('w[n]')
xlabel('

n')
```

Scilab code Exa 3.15

DTFS:Periodic Convolution Property

```
//clear//
//Example3.15:DTFS: Periodic Convolution Property
clear;
clc;
close;
x = [1,1,0,0,0,0,1];
X = fft(x);
W = X.*X;
w = ifft(W);
w = abs(w);
for i =1:length(x)
  if (abs(w(i))<=0.1)
    w(i) = 0;
  end
end
w = [w($:-1:1) w(2:$)];
N = length(x);
```

```
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot2d3('gnn',[-(N-1):0,1:N-1],w,5)
poly1 = a.children(1).children(1);
poly1.thickness = 3;
title('w[n]')
xlabel('

    n')
```

Chapter 7

The Cotntinuous Time Fourier Transform

Chapter 8

The Cotntinuous Time Fourier Transform

Scilab code Exa 4.1 clear

```
//clear//
//Example 4.1:Continuous Time Fourier Transform of a
//Continuous Time Signal x(t)= exp(−A∗t)u(t), t>0
clear;
clc;
close;
// Analog Signal
A =1;     //Amplitude
Dt = 0.005;
t = 0:Dt:10;
xt = exp(-A*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
figure
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
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')
```

Scilab code Exa 4.1

clear

```
//clear//
//Example 4.1:Continuous Time Fourier Transform of a
//Continuous Time Signal x(t)= exp(−A∗t)u(t), t>0
clear;
clc;
close;
// Analog Signal
A =1;     //Amplitude
Dt = 0.005;
t = 0:Dt:10;
xt = exp(-A*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
figure
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
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')
```

Scilab code Exa 4.2 clear

```
//clear//
//Example 4.2:Continuous Time Fourier Transform of a
//Continuous Time Signal x(t)= exp(-A*abs(t))
clear;
clc;
close;
// Analog Signal
A =1;     //Amplitude
Dt = 0.005;
```

```
t = -4.5:Dt:4.5;
xt = exp(-A*abs(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 = real(XW);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from −
   Wmax to Wmax
XW = [mtlb_fliplr(XW), XW(2:1001)];
subplot(1,1,1)
subplot(2,1,1);
a = gca();
a.y_location = "origin";
plot(t,xt);
xlabel('t in sec.');
ylabel('x(t)')
title('Continuous Time Signal')
subplot(2,1,2);
a = gca();
a.y_location = "origin";
plot(W,XW);
xlabel('Frequency in Radians/Seconds W');
ylabel('X(jW)')
title('Continuous−time Fourier Transform')
```

Scilab code Exa 4.2

clear

```
//clear//
//Example 4.2:Continuous Time Fourier Transform of a
//Continuous Time Signal x(t)= exp(−A*abs(t))
clear;
clc;
```

```
close;
// Analog Signal
A =1;     //Amplitude
Dt = 0.005;
t = -4.5:Dt:4.5;
xt = exp(-A*abs(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 = real(XW);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
XW = [mtlb_fliplr(XW), XW(2:1001)];
subplot(1,1,1)
subplot(2,1,1);
a = gca();
a.y_location = "origin";
plot(t,xt);
xlabel('t in sec.');
ylabel('x(t)')
title('Continuous Time Signal')
subplot(2,1,2);
a = gca();
a.y_location = "origin";
plot(W,XW);
xlabel('Frequency in Radians/Seconds W');
ylabel('X(jW)')
title('Continuous-time Fourier Transform')
```

Scilab code Exa 4.4 clear

```
//clear//
//Example 4.4
// Continuous Time Fourier Transform
//and Frequency Response of a Square Waveform
// x(t)= A, from -T1 to T1
clear;
clc;
close;
// CTS Signal
A =1;    //Amplitude
Dt = 0.005;
T1 = 4;  //Time in seconds
t = -T1/2:Dt:T1/2;
for i = 1:length(t)
  xt(i) = A;
end
//
// Continuous-time Fourier Transform
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)];
//
subplot(2,1,1);
a = gca();
a.data_bounds=[-4,0;4,2];
a.y_location ="origin";
plot(t,xt);
xlabel('t in msec.');
title('Contiuous Time Signal x(t)')
```

```
subplot(2,1,2);
a = gca();
a.y_location ="origin";
plot(W,XW_Mag);
xlabel('Frequency in Radians/Seconds');
title('Continuous−time Fourier Transform  X(jW)')
```

Scilab code Exa 4.4

clear

```
//clear//
//Example 4.4
// Continuous Time Fourier Transform
//and Frequency Response of a Square Waveform
// x(t)= A, from −T1 to T1
clear;
clc;
close;
// CTS Signal
A =1;     //Amplitude
Dt = 0.005;
T1 = 4;   //Time in seconds
t = -T1/2:Dt:T1/2;
for i = 1:length(t)
  xt(i) = A;
end
//
// Continuous−time Fourier Transform
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)];
//
subplot(2,1,1);
a = gca();
a.data_bounds=[-4,0;4,2];
a.y_location ="origin";
plot(t,xt);
xlabel('t in msec.');
title('Contiuous Time Signal x(t)')
subplot(2,1,2);
a = gca();
a.y_location ="origin";
plot(W,XW_Mag);
xlabel('Frequency in Radians/Seconds');
title('Continuous-time Fourier Transform  X(jW)')
```

Scilab code Exa 4.5 clear

```
//clear//
//Example 4.5
// Inverse Continuous Time Fourier Transform
// X(jW)= 1, from -T1 to T1
clear;
clc;
close;
// CTFT
A =1;     //Amplitude
Dw = 0.005;
W1 = 4;   //Time in seconds
w = -W1/2:Dw:W1/2;
for i = 1:length(w)
  XW(i) = A;
end
```

```
XW = XW';
//
//Inverse Continuous−time Fourier Transform
t = -%pi:%pi/length(w):%pi;
xt =(1/(2*%pi))*XW *exp(sqrt(-1)*w'*t)*Dw;
xt = real(xt);
figure
a = gca();
a.y_location =" origin";
a.x_location =" origin";
plot(t,xt);
xlabel('                                    t time
   in Seconds');
title('Inverse Continuous Time Fourier Transform x(t
   )')
```

Scilab code Exa 4.5

clear

```
//clear//
//Example 4.5
// Inverse Continuous Time Fourier Transform
// X(jW)= 1, from −T1 to T1
clear;
clc;
close;
// CTFT
A =1;     //Amplitude
Dw = 0.005;
W1 = 4;  //Time in seconds
w = -W1/2:Dw:W1/2;
for i = 1:length(w)
  XW(i) = A;
end
XW = XW';
//
//Inverse Continuous−time Fourier Transform
```

```
t = -%pi:%pi/length(w):%pi;
xt =(1/(2*%pi))*XW *exp(sqrt(-1)*w'*t)*Dw;
xt = real(xt);
figure
a = gca();
a.y_location =" origin";
a.x_location =" origin";
plot(t,xt);
xlabel('                                         t time
   in Seconds');
title('Inverse Continuous Time Fourier Transform x(t
   )')
```

Scilab code Exa 4.6 clear

```
//clear//
//Example 4.6
// Continuous Time Fourier Transform of Symmetric
// periodic Square waveform
clear;
clc;
close;
// CTFT
T1 = 2;
T = 4*T1;
Wo = 2*%pi/T;
W = -%pi:Wo:%pi;
delta = ones(1,length(W));
XW(1) = (2*%pi*Wo*T1/%pi);
mid_value = ceil(length(W)/2);
for k = 2:mid_value
  XW(k) = (2*%pi*sin((k-1)*Wo*T1)/(%pi*(k-1)));
end
```

```
figure
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',W(mid_value:$),XW,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',W(1:mid_value-1),XW($:-1:2),2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('W in radians/Seconds');
title('Continuous Time Fourier Transform of Periodic
    Square Wave')
```

Scilab code Exa 4.6

clear

```
//clear//
//Example 4.6
// Continuous Time Fourier Transform of Symmetric
// periodic Square waveform
clear;
clc;
close;
// CTFT
T1 = 2;
T = 4*T1;
Wo = 2*%pi/T;
W = -%pi:Wo:%pi;
delta = ones(1,length(W));
XW(1) = (2*%pi*Wo*T1/%pi);
mid_value = ceil(length(W)/2);
for k = 2:mid_value
  XW(k) = (2*%pi*sin((k-1)*Wo*T1)/(%pi*(k-1)));
end
figure
a = gca();
```

```
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',W(mid_value:$),XW,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
plot2d3('gnn',W(1:mid_value-1),XW($:-1:2),2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('W in radians/Seconds');
title('Continuous Time Fourier Transform of Periodic
    Square Wave')
```

Scilab code Exa 4.7 clear

```
//clear//
//Example 4.7
// Continuous Time Fourier Transforms of
// Sinusoidal waveforms (a)sin(Wot) (b)cos(Wot)
clear;
clc;
close;
// CTFT
T1 = 2;
T = 4*T1;
Wo = 2*%pi/T;
W = [-Wo,0,Wo];
ak = (2*%pi*Wo*T1/%pi)/sqrt(-1);
XW = [-ak,0,ak];
ak1 = (2*%pi*Wo*T1/%pi);
XW1 =[ak1,0,ak1];
//
figure
a = gca();
```

```
a.y_location =”origin”;
a.x_location =”origin”;
plot2d3('gnn',W,imag(XW),2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('

    W');
title('CTFT of sin(Wot)')
//
figure
a = gca();
a.y_location =”origin”;
a.x_location =”origin”;
plot2d3('gnn',W,XW1,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('

    W');
title('CTFT of cos(Wot)')
```

Scilab code Exa 4.7

clear

```
//clear//
//Example 4.7
// Continuous Time Fourier Transforms of
// Sinusoidal waveforms (a)sin(Wot) (b)cos(Wot)
clear;
clc;
close;
// CTFT
T1 = 2;
T = 4*T1;
Wo = 2*%pi/T;
W = [-Wo,0,Wo];
```

```
ak = (2*%pi*Wo*T1/%pi)/sqrt(-1);
XW = [-ak,0,ak];
ak1 = (2*%pi*Wo*T1/%pi);
XW1 =[ak1,0,ak1];
//
figure
a = gca();
a.y_location =" origin";
a.x_location =" origin";
plot2d3('gnn',W,imag(XW),2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('

   W');
title('CTFT of sin(Wot)')
//
figure
a = gca();
a.y_location =" origin";
a.x_location =" origin";
plot2d3('gnn',W,XW1,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('

   W');
title('CTFT of cos(Wot)')
```

Scilab code Exa 4.8 clear

```
//clear//
//Example 4.8
```

```
// Continuous Time Fourier Transforms of
// Periodic Impulse Train
clear;
clc;
close;
// CTFT
T = -4:4;;
T1 = 1; //Sampling Interval
xt = ones(1,length(T));
ak = 1/T1;
XW = 2*%pi*ak*ones(1,length(T));
Wo = 2*%pi/T1;
W = Wo*T;
figure
subplot(2,1,1)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',T,xt,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('

    t');
title('Periodic Impulse Train')
subplot(2,1,2)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',W,XW,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('

    t');
title('CTFT of Periodic Impulse Train')
```

Scilab code Exa 4.8 clear

```
//clear//
//Example 4.8
// Continuous Time Fourier Transforms of
// Periodic Impulse Train
clear;
clc;
close;
// CTFT
T = -4:4;;
T1 = 1; //Sampling Interval
xt = ones(1,length(T));
ak = 1/T1;
XW = 2*%pi*ak*ones(1,length(T));
Wo = 2*%pi/T1;
W = Wo*T;
figure
subplot(2,1,1)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',T,xt,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('

    t');
title('Periodic Impulse Train')
subplot(2,1,2)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',W,XW,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('

    t');
title('CTFT of Periodic Impulse Train')
```

Scilab code Exa 4.9 clear

```
//clear//
//Example 4.9:Continuous Time Fourier Transform
   Properties:
//Linearity and Time Shift Property
clear;
clc;
close;
// CTFT
t1 = -1/2:0.1:1/2;
t2 = -3/2:0.1:3/2;
x1 = ones(1,length(t1));
x2 = ones(1,length(t2));
t3 = t1+2.5;
t4 = t2+2.5;
x1 = (1/2)*x1;
x = [x2(1:floor(length(x2)/3)),x1+x2(ceil(length(x2)
   /3):$-floor(length(x2)/3)),x2(($-ceil(length(x2)
   /3))+2:$)];
subplot(3,1,1)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot(t1,x1)
xtitle('x1(t)')
subplot(3,1,2)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot(t2,x2)
xtitle('x2(t)')
```

```
subplot(3,1,3)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot(t4,x)
xtitle('x(t)')
```

Scilab code Exa 4.9

clear

```
//clear//
//Example 4.9:Continuous Time Fourier Transform
    Properties:
//Linearity and Time Shift Property
clear;
clc;
close;
// CTFT
t1 = -1/2:0.1:1/2;
t2 = -3/2:0.1:3/2;
x1 = ones(1,length(t1));
x2 = ones(1,length(t2));
t3 = t1+2.5;
t4 = t2+2.5;
x1 = (1/2)*x1;
x = [x2(1:floor(length(x2)/3)),x1+x2(ceil(length(x2)
   /3):$-floor(length(x2)/3)),x2(($-ceil(length(x2)
   /3))+2:$)];
subplot(3,1,1)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot(t1,x1)
xtitle('x1(t)')
subplot(3,1,2)
a = gca();
a.x_location = "origin";
```

```
a.y_location = "origin";
plot(t2,x2)
xtitle('x2(t)')
subplot(3,1,3)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot(t4,x)
xtitle('x(t)')
```

Scilab code Exa 4.12 clear

```
//clear//
//Example 4.12:Continuous Time Fourier Transform:
//Derivative property
clear;
clc;
close;
// CTFT
t = -1:0.1:1;
x1 = ones(1,length(t));
x2 = [-1,zeros(1,length(t)-2),-1];
x = t;
//differentiation of x can be expressed as
//summation of x1 and x2
subplot(3,1,1)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot(t,x1)
xtitle('x1(t)')
subplot(3,1,2)
a = gca();
```

```
a.x_location = "origin";
a.y_location = "origin";
plot2d3('gnn',t,x2)
xtitle('x2(t)')
subplot(3,1,3)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot(t,x)
xtitle('x(t)')
```

Scilab code Exa 4.12 clear

```
//clear//
//Example 4.12:Continuous Time Fourier Transform:
//Derivative property
clear;
clc;
close;
// CTFT
t = -1:0.1:1;
x1 = ones(1,length(t));
x2 = [-1,zeros(1,length(t)-2),-1];
x = t;
//differentiation of x can be expressed as
//summation of x1 and x2
subplot(3,1,1)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot(t,x1)
xtitle('x1(t)')
subplot(3,1,2)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot2d3('gnn',t,x2)
xtitle('x2(t)')
```

```
subplot(3,1,3)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot(t,x)
xtitle('x(t)')
```

Scilab code Exa 4.18 clear

```
//clear//
//Example 4.18:Frequency Response of Ideal Low pass
   Filter
// X(jW)= 1, from −T1 to T1
clear;
clc;
close;
Wc = 10;     //1 rad/sec
W  = -Wc:0.1:Wc; //Passband of filter
HW0 = 1; //Magnitude of Filter
HW = HW0*ones(1,length(W));
//Inverse Continuous−time Fourier Transform
t = -%pi:%pi/length(W):%pi;
Dw = 0.1;
ht =(1/(2*%pi))*HW *exp(sqrt(-1)*W'*t)*Dw;
ht = real(ht);
figure
subplot(2,1,1)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot(W,HW);
xtitle('Frequency Response of Filter H(jW)')
subplot(2,1,2)
```

```
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot(t,ht);
xtitle('Impulse Response of Filter h(t)')
```

Scilab code Exa 4.18 clear

```
//clear//
//Example 4.18:Frequency Response of Ideal Low pass
   Filter
// X(jW)= 1, from -T1 to T1
clear;
clc;
close;
Wc = 10;     //1 rad/sec
W  = -Wc:0.1:Wc; //Passband of filter
HW0 = 1; //Magnitude of Filter
HW = HW0*ones(1,length(W));
//Inverse Continuous-time Fourier Transform
t = -%pi:%pi/length(W):%pi;
Dw = 0.1;
ht =(1/(2*%pi))*HW *exp(sqrt(-1)*W'*t)*Dw;
ht = real(ht);
figure
subplot(2,1,1)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot(W,HW);
xtitle('Frequency Response of Filter H(jW)')
subplot(2,1,2)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot(t,ht);
xtitle('Impulse Response of Filter h(t)')
```

Scilab code Exa 4.22 clear

```
//clear//
//Figure 4.22
//Plotting Continuous Time Fourier Transform of
//Impulse Response h(t)= exp(-A*t)u(t), t>0
clear;
clc;
close;
// Analog Signal
A =1;     //Amplitude
Dt = 0.005;
t = 0:Dt:10;
ht = exp(-A*t);
// Continuous-time Fourier Transform
Wmax = 2*%pi*1;         //Analog Frequency = 1Hz
K = 4;
k = 0:(K/1000):K;
W = k*Wmax/K;
HW = ht* exp(-sqrt(-1)*t'*W) * Dt;
HW_Mag = abs(HW);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -
   Wmax to Wmax
HW_Mag = [mtlb_fliplr(HW_Mag), HW_Mag(2:1001)];
//Plotting Continuous Time Signal
figure
a = gca();
a.y_location = "origin";
plot(t,ht);
xlabel('t in sec.');
title('Impulse Response h(t)')
figure
```

```
//Plotting Magnitude Response of CTS
a = gca();
a.y_location = "origin";
plot(W,HW_Mag);
xlabel('Frequency in Radians/Seconds---> W');
title('Frequency Response H(jW)')
```

Scilab code Exa 4.22

clear

```
//clear//
//Figure 4.22
//Plotting Continuous Time Fourier Transform of
//Impulse Response h(t)= exp(-A*t)u(t), t>0
clear;
clc;
close;
// Analog Signal
A =1;      //Amplitude
Dt = 0.005;
t = 0:Dt:10;
ht = exp(-A*t);
// Continuous-time Fourier Transform
Wmax = 2*%pi*1;          //Analog Frequency = 1Hz
K = 4;
k = 0:(K/1000):K;
W = k*Wmax/K;
HW = ht* exp(-sqrt(-1)*t'*W) * Dt;
HW_Mag = abs(HW);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -
   Wmax to Wmax
HW_Mag = [mtlb_fliplr(HW_Mag), HW_Mag(2:1001)];
//Plotting Continuous Time Signal
figure
a = gca();
a.y_location = "origin";
plot(t,ht);
```

```
xlabel('t in sec.');
title('Impulse Response h(t)')
figure
//Plotting Magnitude Response of CTS
a = gca();
a.y_location = "origin";
plot(W,HW_Mag);
xlabel('Frequency in Radians/Seconds---> W');
title('Frequency Response H(jW)')
```

Scilab code Exa 4.23 clear

```
//Figure 4.23:Multiplication Property of CTFT
clear;
clc;
close;
W1 = -1:0.1:1;
W2 = -2:0.1:2;
W = -3:0.1:3;
//Fourier Transform of sinc funcion is square wave
XW1 = (1/%pi)*ones(1,length(W1)); //CTFT of x1(t)
XW2 = (1/(2*%pi))*ones(1,length(W2));//CTFT of x2(t)
XW = (1/2)*convol(XW1,XW2);//CTFT of x(t)=x1(t)*x2(t
    )
//X(jw) = linear convolution of X1(jw)and X2(jw)
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot(W,XW);
xlabel('Frequency in Radians/Seconds---> W');
title('Multiplication Property X(jW)')
```

Scilab code Exa 4.23

clear

```
//Figure 4.23:Multiplication Property of CTFT
clear;
clc;
close;
W1 = -1:0.1:1;
W2 = -2:0.1:2;
W = -3:0.1:3;
//Fourier Transform of sinc funcion is square wave
XW1 = (1/%pi)*ones(1,length(W1)); //CTFT of x1(t)
XW2 = (1/(2*%pi))*ones(1,length(W2));//CTFT of x2(t)
XW = (1/2)*convol(XW1,XW2);//CTFT of x(t)=x1(t)*x2(t
   )
//X(jw) = linear convolution of X1(jw)and X2(jw)
figure
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot(W,XW);
xlabel('Frequency in Radians/Seconds---> W');
title('Multiplication Property X(jW)')
```

Chapter 9

The Discreet Time Fourier Transform

Chapter 10

The Discreet Time Fourier Transform

Scilab code Exa 5.1 Discrete Time Fourier Transform of discrete sequence

```
//clear//
//Example 5.1:Discrete Time Fourier Transform of
   discrete sequence
//x[n]= (a^n).u[n], a>0 and a<0
clear;
clc;
close;
// DTS Signal
a1 = 0.5;
a2 = -0.5;
max_limit = 10;
for n = 0:max_limit-1
  x1(n+1) = (a1^n);
  x2(n+1) = (a2^n);
end
n = 0:max_limit-1;
```

```
// Discrete-time Fourier Transform
Wmax = 2*%pi;
K = 4;
k = 0:(K/1000):K;
W = k*Wmax/K;
x1 = x1';
x2 = x2';
XW1 = x1* exp(-sqrt(-1)*n'*W);
XW2 = x2* exp(-sqrt(-1)*n'*W);
XW1_Mag = abs(XW1);
XW2_Mag = abs(XW2);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
XW1_Mag = [mtlb_fliplr(XW1_Mag), XW1_Mag(2:1001)];
XW2_Mag = [mtlb_fliplr(XW2_Mag), XW2_Mag(2:1001)];
[XW1_Phase,db] = phasemag(XW1);
[XW2_Phase,db] = phasemag(XW2);
XW1_Phase = [-mtlb_fliplr(XW1_Phase),XW1_Phase
    (2:1001)];
XW2_Phase = [-mtlb_fliplr(XW2_Phase),XW2_Phase
    (2:1001)];
//plot for a>0
figure
subplot(3,1,1);
plot2d3('gnn',n,x1);
xtitle('Discrete Time Sequence x[n] for a>0')
subplot(3,1,2);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW1_Mag);
title('Magnitude Response abs(X(jW))')
subplot(3,1,3);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW1_Phase);
title('Phase Response <(X(jW))')
```

```
//plot for a<0
figure
subplot(3,1,1);
plot2d3('gnn',n,x2);
xtitle('Discrete Time Sequence x[n] for a>0')
subplot(3,1,2);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW2_Mag);
title('Magnitude Response abs(X(jW))')
subplot(3,1,3);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW2_Phase);
title('Phase Response <(X(jW))')
```

Scilab code Exa 5.1

Discrete Time Fourier Transform of discrete sequence

```
//clear//
//Example 5.1:Discrete Time Fourier Transform of
   discrete sequence
//x[n]= (a^n).u[n], a>0 and a<0
clear;
clc;
close;
// DTS Signal
a1 = 0.5;
a2 = -0.5;
max_limit = 10;
for n = 0:max_limit-1
  x1(n+1) = (a1^n);
  x2(n+1) = (a2^n);
end
n = 0:max_limit-1;
```

```
// Discrete-time Fourier Transform
Wmax = 2*%pi;
K = 4;
k = 0:(K/1000):K;
W = k*Wmax/K;
x1 = x1';
x2 = x2';
XW1 = x1* exp(-sqrt(-1)*n'*W);
XW2 = x2* exp(-sqrt(-1)*n'*W);
XW1_Mag = abs(XW1);
XW2_Mag = abs(XW2);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -
    Wmax to Wmax
XW1_Mag = [mtlb_fliplr(XW1_Mag), XW1_Mag(2:1001)];
XW2_Mag = [mtlb_fliplr(XW2_Mag), XW2_Mag(2:1001)];
[XW1_Phase,db] = phasemag(XW1);
[XW2_Phase,db] = phasemag(XW2);
XW1_Phase = [-mtlb_fliplr(XW1_Phase),XW1_Phase
    (2:1001)];
XW2_Phase = [-mtlb_fliplr(XW2_Phase),XW2_Phase
    (2:1001)];
//plot for a>0
figure
subplot(3,1,1);
plot2d3('gnn',n,x1);
xtitle('Discrete Time Sequence x[n] for a>0')
subplot(3,1,2);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW1_Mag);
title('Magnitude Response abs(X(jW))')
subplot(3,1,3);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d(W,XW1_Phase);
title('Phase Response <(X(jW))')
```

```
//plot for a<0
figure
subplot(3,1,1);
plot2d3('gnn',n,x2);
xtitle('Discrete Time Sequence x[n] for a>0')
subplot(3,1,2);
a = gca();
a.y_location =" origin";
a.x_location =" origin";
plot2d(W,XW2_Mag);
title('Magnitude Response abs(X(jW))')
subplot(3,1,3);
a = gca();
a.y_location =" origin";
a.x_location =" origin";
plot2d(W,XW2_Phase);
title('Phase Response <(X(jW))')
```

Scilab code Exa 5.2 Discrete Time Fourier Transform

```
//clear//
//Example 5.2:Discrete Time Fourier Transform of
//x[n]= (a^abs(n)) a>0 and a<0
clear;
clc;
close;
// DTS Signal
a = 0.5;
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)];
//plot for abs(a)<1
figure
subplot(2,1,1);
a = gca();
a.y_location =" origin";
a.x_location =" origin";
plot2d3('gnn',n,x);
xtitle('Discrete Time Sequence x[n] for a>0')
subplot(2,1,2);
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

Discrete Time Fourier Transform

```
//clear//
//Example 5.2:Discrete Time Fourier Transform of
//x[n]= (a^abs(n)) a>0 and a<0
clear;
clc;
close;
// DTS Signal
a = 0.5;
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)];
//plot for abs(a)<1
figure
subplot(2,1,1);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',n,x);
xtitle('Discrete Time Sequence x[n] for a>0')
subplot(2,1,2);
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 5.3 Discrete Time Fourier Transform

```
//clear//
//Example 5.3:Discrete Time Fourier Transform of
//x[n]= 1 , abs(n)<=N1
clear;
clc;
close;
```

```
// DTS Signal
N1 = 2;
n = -N1:N1;
x = ones(1,length(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)];
//plot for abs(a)<1
figure
subplot(2,1,1);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',n,x);
xtitle('Discrete Time Sequence x[n]')
subplot(2,1,2);
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 5.3

Discrete Time Fourier Transform

```
//clear//
//Example 5.3:Discrete Time Fourier Transform of
//x[n]= 1 , abs(n)<=N1
clear;
clc;
```

```
close;
// DTS Signal
N1 = 2;
n = -N1:N1;
x = ones(1,length(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)];
//plot for abs(a)<1
figure
subplot(2,1,1);
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',n,x);
xtitle('Discrete Time Sequence x[n]')
subplot(2,1,2);
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 5.5 Scilab code Exa 5.5 Time Fourier Transform:x[n]= cos(nWo)

```
//clear//
//Example5.5:Discrete Time Fourier Transform:x[n]=
   cos(nWo)
clear;
clc;
close;
N = 5;
Wo = 2*%pi/N;
W = [-Wo,0,Wo];
XW =[%pi,0,%pi];
//
figure
a = gca();
a.y_location =" origin";
a.x_location =" origin";
plot2d3('gnn',W,XW,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('

  W');
title('DTFT of cos(nWo)')
disp(Wo)
```

Time Fourier Transform:x[n]= cos(nWo)

```
//clear//
//Example5.5:Discrete Time Fourier Transform:x[n]=
   cos(nWo)
clear;
clc;
close;
N = 5;
Wo = 2*%pi/N;
W = [-Wo,0,Wo];
XW =[%pi,0,%pi];
//
figure
```

```
a = gca();
a.y_location ="origin";
a.x_location ="origin";
plot2d3('gnn',W,XW,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xlabel('

   W');
title('DTFT of cos(nWo)')
disp(Wo)
```

Scilab code Exa 5.6 Discrete Time Fourier Transform

```
//clear//
//Example5.6:Discrete Time Fourier Transform of
// Periodic Impulse Train
clear;
clc;
close;
N = 5;
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;
figure
subplot(2,1,1)
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')
subplot(2,1,2)
a = gca();
a.y_location ="origin";
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')
disp(Wo)
```

Scilab code Exa 5.6

Discrete Time Fourier Transform

```
//clear//
//Example5.6:Discrete Time Fourier Transform of
// Periodic Impulse Train
clear;
clc;
close;
N = 5;
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;
figure
subplot(2,1,1)
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')
subplot(2,1,2)
a = gca();
a.y_location ="origin";
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')
disp(Wo)
```

Scilab code Exa 5.7 Frequency Shifting Property of DTFT

```
//clear//
//Example 5.7:Frequency Shifting Property of DTFT:
   Frequency Response of Ideal Low pass Filter and
   HPF
```

```
clear;
clc;
close;
Wc = 1;     //1 rad/sec
W  = -Wc:0.1:Wc; //Passband of filter
H0 = 1; //Magnitude of Filter
HlpW = H0*ones(1,length(W));
Whp1 = W+%pi;
Whp2 = -W-%pi;
figure
subplot(2,1,1)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
a.data_bounds=[-%pi,0;%pi,2];
plot2d(W,HlpW);
xtitle('Frequency Response of LPF H(exp(jW))')
subplot(2,1,2)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
a.data_bounds=[-2*%pi,0;2*%pi,2];
plot2d(Whp1,HlpW);
plot2d(Whp2,HlpW);
xtitle('Frequency Response of HPF H(exp(jW))')
```

Scilab code Exa 5.7

Frequency Shifting Property of DTFT

```
//clear//
//Example 5.7:Frequency Shifting Property of DTFT:
   Frequency Response of Ideal Low pass Filter and
   HPF
clear;
clc;
close;
Wc = 1;     //1 rad/sec
```

```
W  = -Wc:0.1:Wc; //Passband of filter
H0 = 1; //Magnitude of Filter
HlpW = H0*ones(1,length(W));
Whp1 = W+%pi;
Whp2 = -W-%pi;
figure
subplot(2,1,1)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
a.data_bounds=[-%pi,0;%pi,2];
plot2d(W,HlpW);
xtitle('Frequency Response of LPF H(exp(jW))')
subplot(2,1,2)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
a.data_bounds=[-2*%pi,0;2*%pi,2];
plot2d(Whp1,HlpW);
plot2d(Whp2,HlpW);
xtitle('Frequency Response of HPF H(exp(jW))')
```

Scilab code Exa 5.9 Expansion Property of DTFT

```
//clear//
//Example 5.9:Time Expansion Property of DTFT
clear;
close;
clc;
n = -1:11;
x = [0,1,2,1,2,1,2,1,2,1,2,0,0];
y = [1,1,1,1,1];
y_2_n =zeros(1,2*length(y)+1);
```

```
y_2_n(1:2:2*length(y)) = y;
y_2_n = [0 y_2_n 0];
y_2_n_1 = [0,y_2_n(1:$-1)];
x_r = y_2_n+2*y_2_n_1;
y = [0,y,zeros(1,7)];
figure
subplot(4,1,1)
plot2d3('gnn',n,y)
title('y[n]')
subplot(4,1,2)
plot2d3('gnn',n,y_2_n)
title('y(2)[n]')
subplot(4,1,3)
plot2d3('gnn',n,y_2_n_1)
title('y(2)[n-1]')
subplot(4,1,4)
plot2d3('gnn',n,x)
title('x[n]=y(2)[n]+2*y(2)[n-1]')
```

Scilab code Exa 5.9

Expansion Property of DTFT

```
//clear//
//Example 5.9:Time Expansion Property of DTFT
clear;
close;
clc;
n = -1:11;
x = [0,1,2,1,2,1,2,1,2,1,2,0,0];
y = [1,1,1,1,1];
y_2_n =zeros(1,2*length(y)+1);
y_2_n(1:2:2*length(y)) = y;
y_2_n = [0 y_2_n 0];
y_2_n_1 = [0,y_2_n(1:$-1)];
x_r = y_2_n+2*y_2_n_1;
y = [0,y,zeros(1,7)];
figure
```

```
subplot(4,1,1)
plot2d3('gnn',n,y)
title('y[n]')
subplot(4,1,2)
plot2d3('gnn',n,y_2_n)
title('y(2)[n]')
subplot(4,1,3)
plot2d3('gnn',n,y_2_n_1)
title('y(2)[n-1]')
subplot(4,1,4)
plot2d3('gnn',n,x)
title('x[n]=y(2)[n]+2*y(2)[n-1]')
```

Scilab code Exa 5.12 IDTFT:Impulse Response of Ideal Low pass Filter

```
//clear//
//Example 5.12:IDTFT:Impulse Response of Ideal Low
   pass Filter
clear;
clc;
close;
Wc = 1;     //1 rad/sec
W  = -Wc:0.1:Wc; //Passband of filter
H0 = 1; //Magnitude of Filter
HlpW = H0*ones(1,length(W));
//Inverse Discrete-time Fourier Transform
t = -2*%pi:2*%pi/length(W):2*%pi;
ht =(1/(2*%pi))*HlpW *exp(sqrt(-1)*W'*t);
ht = real(ht);
figure
subplot(2,1,1)
a = gca();
a.y_location ="origin";
```

```
a.x_location =" origin";
a.data_bounds=[-%pi,0;%pi,2];
plot2d(W,HlpW,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xtitle('Frequency Response of LPF H(exp(jW))')
subplot(2,1,2)
a = gca();
a.y_location =" origin";
a.x_location =" origin";
a.data_bounds=[-2*%pi,-1;2*%pi,2];
plot2d3('gnn',t,ht);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xtitle('Impulse Response of LPF h(t)')
```

Scilab code Exa 5.12

IDTFT:Impulse Response of Ideal Low pass Filter

```
//clear//
//Example 5.12:IDTFT:Impulse Response of Ideal Low
   pass Filter
clear;
clc;
close;
Wc = 1;     //1 rad/sec
W  = -Wc:0.1:Wc; //Passband of filter
H0 = 1; //Magnitude of Filter
HlpW = H0*ones(1,length(W));
//Inverse Discrete-time Fourier Transform
t = -2*%pi:2*%pi/length(W):2*%pi;
ht =(1/(2*%pi))*HlpW *exp(sqrt(-1)*W'*t);
ht = real(ht);
figure
subplot(2,1,1)
a = gca();
a.y_location =" origin";
```

```
a.x_location =" origin";
a.data_bounds=[-%pi,0;%pi,2];
plot2d(W,HlpW,2);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xtitle('Frequency Response of LPF H(exp(jW))')
subplot(2,1,2)
a = gca();
a.y_location =" origin";
a.x_location =" origin";
a.data_bounds=[-2*%pi,-1;2*%pi,2];
plot2d3('gnn',t,ht);
poly1 = a.children(1).children(1);
poly1.thickness = 3;
xtitle('Impulse Response of LPF h(t)')
```

Scilab code Exa 5.15 Multiplication Property of DTFT

```
//clear//
//Example5.15:Multiplication Property of DTFT
clear;
clc;
close;
n = 1:100;
x2 = [3/4,sin(0.75*%pi*n)./(%pi*n)];
x1 = [1/2,sin(0.5*%pi*n)./(%pi*n)];
x = x1.*x2;
Wmax = %pi;
K = 1;
k = 0:(K/1000):K;
W = k*Wmax/K;
 n = 0:100;
XW1 = x1* exp(-sqrt(-1)*n'*W);
```

```
XW2 = x2* exp(-sqrt(-1)*n'*W);
XW = x* exp(-sqrt(-1)*n'*W);
XW1_Mag = real(XW1);
XW2_Mag = real(XW2);
XW_Mag = real(XW);
W = [-mtlb_fliplr(W), W(2:$)]; // Omega from -Wmax
    to Wmax
XW1_Mag = [mtlb_fliplr(XW1_Mag), XW1_Mag(2:$)];
XW2_Mag = [mtlb_fliplr(XW2_Mag), XW2_Mag(2:$)];
XW_Mag = [mtlb_fliplr(XW_Mag), XW_Mag(2:$)];
figure
subplot(3,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot(W,XW1_Mag);
title('DTFT X1(exp(jW))');
subplot(3,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot(W,XW2_Mag);
title('DTFT X2(exp(jW))');
subplot(3,1,3)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot(W,XW_Mag);
title('Multiplication Property of DTFT');
```

Scilab code Exa 5.15

Multiplication Property of DTFT

```
//clear//
//Example5.15:Multiplication Property of DTFT
clear;
clc;
```

```
close;
n = 1:100;
x2 = [3/4,sin(0.75*%pi*n)./(%pi*n)];
x1 = [1/2,sin(0.5*%pi*n)./(%pi*n)];
x = x1.*x2;
Wmax = %pi;
K = 1;
k = 0:(K/1000):K;
W = k*Wmax/K;
 n = 0:100;
XW1 = x1* exp(-sqrt(-1)*n'*W);
XW2 = x2* exp(-sqrt(-1)*n'*W);
XW = x* exp(-sqrt(-1)*n'*W);
XW1_Mag = real(XW1);
XW2_Mag = real(XW2);
XW_Mag = real(XW);
W = [-mtlb_fliplr(W), W(2:$)]; // Omega from -Wmax
   to Wmax
XW1_Mag = [mtlb_fliplr(XW1_Mag), XW1_Mag(2:$)];
XW2_Mag = [mtlb_fliplr(XW2_Mag), XW2_Mag(2:$)];
XW_Mag = [mtlb_fliplr(XW_Mag), XW_Mag(2:$)];
figure
subplot(3,1,1)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot(W,XW1_Mag);
title('DTFT X1(exp(jW))');
subplot(3,1,2)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
plot(W,XW2_Mag);
title('DTFT X2(exp(jW))');
subplot(3,1,3)
a = gca();
a.y_location = "origin";
a.x_location = "origin";
```

```
plot(W,XW_Mag);
title('Multiplication Property of DTFT');
```

Chapter 11

Time and Frequency Characterization of Signals and Systems

Chapter 12

Time and Frequency Characterization of Signals and Systems

Scilab code Exa 6.1 Phase Response and Group Delay

```
//clear//
//Example6.1:Phase Response and Group Delay
clear;
clc;
close;
f1 = 50;
f2 = 150;
f3 = 300;
w1 = 315;
tuo1 = 0.066;
w2 = 943;
tuo2 = 0.033;
w3 = 1888;
tuo3 = 0.058;
f = 0:0.1:400;
```

```
W = 2*%pi*f;
for i =1:length(f)
 num1(i) = (1+(sqrt(-1)*f(i)/f1)^2-2*sqrt(-1)*tuo1*(
    f(i)/f1));
 den1(i) = (1+(sqrt(-1)*f(i)/f1)^2+2*sqrt(-1)*tuo1*(
    f(i)/f1));
 H1W(i) = num1(i)/den1(i);
 num2(i) = (1+(sqrt(-1)*f(i)/f2)^2-2*sqrt(-1)*tuo2*(
    f(i)/f2));
 den2(i) = (1+(sqrt(-1)*f(i)/f2)^2+2*sqrt(-1)*tuo2*(
    f(i)/f2));
 H2W(i) = num2(i)/den2(i);
 num3(i) = (1+(sqrt(-1)*f(i)/f3)^2-2*sqrt(-1)*tuo3*(
    f(i)/f3));
 den3(i) = (1+(sqrt(-1)*f(i)/f3)^2+2*sqrt(-1)*tuo3*(
    f(i)/f3));
 H3W(i) = num3(i)/den3(i);
 H_W(i) = H1W(i)*H2W(i);
 HW(i) = H_W(i)*H3W(i);
  phase1(i) = -2*atan((2*tuo1*(f(i)/f1))/(1.001-(f(i
    )/f1)^2));
  phase2(i) = -2*atan((2*tuo2*(f(i)/f2))/(1.001-(f(i
    )/f2)^2));
  phase3(i) = -2*atan((2*tuo3*(f(i)/f3))/(1.001-(f(i
    )/f3)^2));
  phase_total(i) = phase1(i)+phase2(i)+phase3(i);
 if(f(i)<=50)
    W_phase1(i) = -2*atan((2*tuo1*(f(i)/f1))
       /(1.001-(f(i)/f1)^2));
    W_phase2(i) = -2*atan((2*tuo2*(f(i)/f2))
       /(1.001-(f(i)/f2)^2));
    W_phase3(i) = -2*atan((2*tuo3*(f(i)/f3))
       /(1.001-(f(i)/f3)^2));
    group_delay(i) = -phase_total(i)*0.1/%pi;   //
       delta_f= 0.1
 elseif(f(i)>=50 & f(i)<=150)
    W_phase1(i)= -2*%pi-2*atan((2*tuo1*(f(i)/f1))
       /(1.001-(f(i)/f1)^2));
    W_phase2(i)= -2*atan((2*tuo2*(f(i)/f2))/(1.001-(
       f(i)/f2)^2));
    W_phase3(i)= -2*atan((2*tuo3*(f(i)/f3))/(1.001-(
       f(i)/f3)^2));
    group_delay(i) = -phase_total(i)*0.1/(2*%pi);
 elseif(f(i)>=150 & f(i)<=300)
    W_phase1(i)= -2*atan((2*tuo1*(f(i)/f1))/(1.001-(
       f(i)/f1)^2));
    W_phase2(i)= -4*%pi-2*atan((2*tuo2*(f(i)/f2))
       /(1.001-(f(i)/f2)^2));
    W_phase3(i)= -2*atan((2*tuo3*(f(i)/f3))/(1.001-(
       f(i)/f3)^2));
    group_delay(i) = -phase_total(i)*0.1/(4*%pi);
 elseif(f(i)>300 & f(i)<=400)
    W_phase1(i)= -2*atan((2*tuo1*(f(i)/f1))/(1.001-(
       f(i)/f1)^2));
    W_phase2(i)= -2*atan((2*tuo2*(f(i)/f2))/(1.001-(
       f(i)/f2)^2));
    W_phase3(i)= -6*%pi-2*atan((2*tuo3*(f(i)/f3))
       /(1.001-(f(i)/f3)^2));
    group_delay(i) = -phase_total(i)*0.1/(4*%pi);
 end
 if(f(i)==300.1)
   W_phase_total(i) = 2*%pi+W_phase1(i)+W_phase2(i)+
      W_phase3(i);
 else
   W_phase_total(i) = W_phase1(i)+W_phase2(i)+
      W_phase3(i);
 end
end
figure
plot2d(f,phase_total,2)
xtitle('Principal phase','Frequency(Hz)','Phase(rad)
   ');
figure
plot2d(f,W_phase_total,2)
xtitle('unwrapped phase','Frequency(Hz)','Phase(rad)
   ');
```

```
figure
plot2d(f,abs(group_delay),2)
xtitle('group delay','Frequency(Hz)','Group Delay(
    sec)');
```

Scilab code Exa 6.1

Phase Response and Group Delay

```
//clear//
//Example6.1:Phase Response and Group Delay
clear;
clc;
close;
f1 = 50;
f2 = 150;
f3 = 300;
w1 = 315;
tuo1 = 0.066;
w2 = 943;
tuo2 = 0.033;
w3 = 1888;
tuo3 = 0.058;
f = 0:0.1:400;
W = 2*%pi*f;
for i =1:length(f)
 num1(i) = (1+(sqrt(-1)*f(i)/f1)^2-2*sqrt(-1)*tuo1*(
    f(i)/f1));
 den1(i) = (1+(sqrt(-1)*f(i)/f1)^2+2*sqrt(-1)*tuo1*(
    f(i)/f1));
 H1W(i) = num1(i)/den1(i);
 num2(i) = (1+(sqrt(-1)*f(i)/f2)^2-2*sqrt(-1)*tuo2*(
    f(i)/f2));
 den2(i) = (1+(sqrt(-1)*f(i)/f2)^2+2*sqrt(-1)*tuo2*(
    f(i)/f2));
 H2W(i) = num2(i)/den2(i);
 num3(i) = (1+(sqrt(-1)*f(i)/f3)^2-2*sqrt(-1)*tuo3*(
    f(i)/f3));
```

```
den3(i) = (1+(sqrt(-1)*f(i)/f3)^2+2*sqrt(-1)*tuo3*(
   f(i)/f3));
H3W(i) = num3(i)/den3(i);
H_W(i) = H1W(i)*H2W(i);
HW(i) = H_W(i)*H3W(i);
 phase1(i) = -2*atan((2*tuo1*(f(i)/f1))/(1.001-(f(i
    )/f1)^2));
 phase2(i) = -2*atan((2*tuo2*(f(i)/f2))/(1.001-(f(i
    )/f2)^2));
 phase3(i) = -2*atan((2*tuo3*(f(i)/f3))/(1.001-(f(i
    )/f3)^2));
 phase_total(i) = phase1(i)+phase2(i)+phase3(i);
if(f(i)<=50)
   W_phase1(i) = -2*atan((2*tuo1*(f(i)/f1))
      /(1.001-(f(i)/f1)^2));
   W_phase2(i) = -2*atan((2*tuo2*(f(i)/f2))
      /(1.001-(f(i)/f2)^2));
   W_phase3(i) = -2*atan((2*tuo3*(f(i)/f3))
      /(1.001-(f(i)/f3)^2));
   group_delay(i) = -phase_total(i)*0.1/%pi;  //
      delta_f= 0.1
elseif(f(i)>=50 & f(i)<=150)
   W_phase1(i)= -2*%pi-2*atan((2*tuo1*(f(i)/f1))
      /(1.001-(f(i)/f1)^2));
   W_phase2(i)= -2*atan((2*tuo2*(f(i)/f2))/(1.001-(
      f(i)/f2)^2));
   W_phase3(i)= -2*atan((2*tuo3*(f(i)/f3))/(1.001-(
      f(i)/f3)^2));
   group_delay(i) = -phase_total(i)*0.1/(2*%pi);
elseif(f(i)>=150 & f(i)<=300)
   W_phase1(i)= -2*atan((2*tuo1*(f(i)/f1))/(1.001-(
      f(i)/f1)^2));
   W_phase2(i)= -4*%pi-2*atan((2*tuo2*(f(i)/f2))
      /(1.001-(f(i)/f2)^2));
   W_phase3(i)= -2*atan((2*tuo3*(f(i)/f3))/(1.001-(
      f(i)/f3)^2));
   group_delay(i) = -phase_total(i)*0.1/(4*%pi);
elseif(f(i)>300 & f(i)<=400)
```

```
    W_phase1(i)= -2*atan((2*tuo1*(f(i)/f1))/(1.001-(
        f(i)/f1)^2));
    W_phase2(i)= -2*atan((2*tuo2*(f(i)/f2))/(1.001-(
        f(i)/f2)^2));
    W_phase3(i)= -6*%pi-2*atan((2*tuo3*(f(i)/f3))
        /(1.001-(f(i)/f3)^2));
    group_delay(i) = -phase_total(i)*0.1/(4*%pi);
 end
 if(f(i)==300.1)
   W_phase_total(i) = 2*%pi+W_phase1(i)+W_phase2(i)+
       W_phase3(i);
 else
   W_phase_total(i) = W_phase1(i)+W_phase2(i)+
       W_phase3(i);
 end
end
figure
plot2d(f,phase_total,2)
xtitle('Principal phase','Frequency(Hz)','Phase(rad)
   ');
figure
plot2d(f,W_phase_total,2)
xtitle('unwrapped phase','Frequency(Hz)','Phase(rad)
   ');
figure
plot2d(f,abs(group_delay),2)
xtitle('group delay','Frequency(Hz)','Group Delay(
   sec)');
```

Scilab code Exa 6.3 Analog Lowpass IIR filter design

```
//clear//
//Example6.3:Analog Lowpass IIR filter design
```

```
//Cutoff frequency Fc = 500Hz
//Passband ripple 1-0.05 and stopband ripple = 0.05
clear;
close;
clc;
hs_butt = analpf(5,'butt',[0.05,0.05],500);
hs_ellip = analpf(5,'ellip',[0.05,0.05],500);
fr=0:.1:2000;
hf_butt=freq(hs_butt(2),hs_butt(3),%i*fr);
hm_butt = abs(hf_butt);
hf_ellip=freq(hs_ellip(2),hs_ellip(3),%i*fr);
hm_ellip = abs(hf_ellip);
//Plotting Magnitude Response of Analog IIR Filters
a = gca();
plot2d(fr,hm_butt)
poly1 = a.children(1).children(1);
poly1.foreground = 2;
poly1.thickness = 2;
poly1.line_style = 3;
plot2d(fr,hm_ellip)
poly1 = a.children(1).children(1);
poly1.foreground = 5;
poly1.thickness = 2;
xlabel('Frequency(Hz)')
ylabel('Magnitude of frequency response')
legend(['Butterworth Filter';'Elliptic Filter'])
```

Scilab code Exa 6.3

Analog Lowpass IIR filter design

```
//clear//
//Example6.3:Analog Lowpass IIR filter design
//Cutoff frequency Fc = 500Hz
//Passband ripple 1-0.05 and stopband ripple = 0.05
clear;
close;
clc;
```

```
hs_butt = analpf(5,'butt',[0.05,0.05],500);
hs_ellip = analpf(5,'ellip',[0.05,0.05],500);
fr=0:.1:2000;
hf_butt=freq(hs_butt(2),hs_butt(3),%i*fr);
hm_butt = abs(hf_butt);
hf_ellip=freq(hs_ellip(2),hs_ellip(3),%i*fr);
hm_ellip = abs(hf_ellip);
//Plotting Magnitude Response of Analog IIR Filters
a = gca();
plot2d(fr,hm_butt)
poly1 = a.children(1).children(1);
poly1.foreground = 2;
poly1.thickness = 2;
poly1.line_style = 3;
plot2d(fr,hm_ellip)
poly1 = a.children(1).children(1);
poly1.foreground = 5;
poly1.thickness = 2;
xlabel('Frequency(Hz)')
ylabel('Magnitude of frequency response')
legend(['Butterworth Filter';'Elliptic Filter'])
```

Scilab code Exa 6.4 Bode Plot

```
//clear//
//Example 6.4:Bode Plot
s = %s;
//Open Loop Transfer Function
H = syslin('c',[20000/(s^2+100*s+10000)]);//jw
   replaced by s
clf;
bode(H,0.01,10000)
```

Scilab code Exa 6.4

Bode Plot

```
//clear//
//Example 6.4:Bode Plot
s = %s;
//Open Loop Transfer Function
H = syslin('c',[20000/(s^2+100*s+10000)]);//jw
   replaced by s
clf;
bode(H,0.01,10000)
```

Scilab code Exa 6.5 Bode Plot

```
//clear//
//Example 6.5:Bode Plot
s = %s;
//Open Loop Transfer Function
H = syslin('c',[(100*(1+s))/((10+s)*(100+s))]);//jw
   replaced by s
clf;
bode(H,0.01,10000)
```

Scilab code Exa 6.5 Bode Plot

```
//clear//
//Example 6.5:Bode Plot
s = %s;
//Open Loop Transfer Function
H = syslin('c',[(100*(1+s))/((10+s)*(100+s))]);//jw
   replaced by s
clf;
```

```
bode(H,0.01,10000)
```

Chapter 13

Sampling

Chapter 14

Sampling

Scilab code Exa 7.1 Sinusoidal signal

```
//clear//
//Example7.1:Sinusoidal signal
clear;
close;
clc;
Wm = 2*%pi;
Ws = 2*Wm;
t = -2:0.01:2;
phi = -%pi/2;
x = cos((Ws/2)*t+phi);
y = sin((Ws/2)*t);
subplot(2,1,1)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot(t,x)
title('cos(Ws/2*t+phi)')
subplot(2,1,2)
a = gca();
```

```
a.x_location = "origin";
a.y_location = "origin";
plot(t,y)
title('sin(Ws/2*t)')
```

Scilab code Exa 7.1

Sinusoidal signal

```
//clear//
//Example7.1:Sinusoidal signal
clear;
close;
clc;
Wm = 2*%pi;
Ws = 2*Wm;
t = -2:0.01:2;
phi = -%pi/2;
x = cos((Ws/2)*t+phi);
y = sin((Ws/2)*t);
subplot(2,1,1)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot(t,x)
title('cos(Ws/2*t+phi)')
subplot(2,1,2)
a = gca();
a.x_location = "origin";
a.y_location = "origin";
plot(t,y)
title('sin(Ws/2*t)')
```

Scilab code Exa 7.2 Digital Differentiator

```
//clear//
//Example7.2:Digital Differentiator
syms t n;
T = 0.1; //Sampling time in seconds
xct = sin(%pi*t/T)/(%pi*t);
yct  = diff(xct,t);
disp(yct,'yc(t)=');
t = n*T;
xdn = sin(%pi*t/T)/(%pi*t);
ydn = diff(xdn,n);
disp(ydn,'yd[n]=');
hdn = T*ydn;
disp(hdn,'hd[n]=');
//Result
//yc(t) = (10*cos(31.415927*t)/t)-(0.3183099*sin
    (31.415927*t)/(t^2))
//yd[n]=(10*cos(3.1415927*n)/n)-3.183*sin(3.1415927*
    n)/(n^2)
//hd[n]=(cos(3.1415927*n)/n)-0.3183*sin(3.1415927*n)
    /(n^2)
```

Scilab code Exa 7.2

Digital Differentiator

```
//clear//
//Example7.2:Digital Differentiator
syms t n;
T = 0.1; //Sampling time in seconds
xct = sin(%pi*t/T)/(%pi*t);
yct  = diff(xct,t);
disp(yct,'yc(t)=');
t = n*T;
xdn = sin(%pi*t/T)/(%pi*t);
ydn = diff(xdn,n);
disp(ydn,'yd[n]=');
hdn = T*ydn;
disp(hdn,'hd[n]=');
```

```
//Result
//yc(t) = (10*cos(31.415927*t)/t)-(0.3183099*sin
   (31.415927*t)/(t^2))
//yd[n]=(10*cos(3.1415927*n)/n)-3.183*sin(3.1415927*
   n)/(n^2)
//hd[n]=(cos(3.1415927*n)/n)-0.3183*sin(3.1415927*n)
   /(n^2)
```

Scilab code Exa 7.3 Half Sample Delay system

```
//clear//
//Example7.3:Half Sample Delay system
syms t n T;
//T = 0.1; //Sampling time in seconds
xct = sin(%pi*t/T)/(%pi*t);
t = t-T/2;
yct_del  = sin(%pi*t/T)/(%pi*t);
disp(yct_del,'Output of Half Sample delay system
   continuous =');
t = n*T-T/2;
xdn = sin(%pi*t/T)/(%pi*t);
ydn_del = xdn;
disp(ydn_del,'Output of Half Sample delay system
   discrete =');
hdn = T*ydn_del;
disp(hdn,'Impulse Response of discrete time half
   sample delay system=');
//Result
//Output of Half Sample delay system continuous =
//sin(3.14*(t-T/2)/T)/(3.14*(t-T/2))
//Output of Half Sample delay system discrete =
// sin(3.14*(n*T-T/2)/T)/(3.14*(n*T-T/2))
// Impulse Response of discrete time half sample
```

```
   delay system=
// T*sin(3.14*(n*T-T/2)/T)/(3.14*(n*T-T/2))
```

Scilab code Exa 7.3

Half Sample Delay system

```
//clear//
//Example7.3:Half Sample Delay system
syms t n T;
//T = 0.1; //Sampling time in seconds
xct = sin(%pi*t/T)/(%pi*t);
t = t-T/2;
yct_del  = sin(%pi*t/T)/(%pi*t);
disp(yct_del,'Output of Half Sample delay system
   continuous =');
t = n*T-T/2;
xdn = sin(%pi*t/T)/(%pi*t);
ydn_del = xdn;
disp(ydn_del,'Output of Half Sample delay system
   discrete =');
hdn = T*ydn_del;
disp(hdn,'Impulse Response of discrete time half
   sample delay system=');
//Result
//Output of Half Sample delay system continuous =
//sin(3.14*(t-T/2)/T)/(3.14*(t-T/2))
//Output of Half Sample delay system discrete =
// sin(3.14*(n*T-T/2)/T)/(3.14*(n*T-T/2))
// Impulse Response of discrete time half sample
   delay system=
// T*sin(3.14*(n*T-T/2)/T)/(3.14*(n*T-T/2))
```

Scilab code Exa 7.4 Period of the sampled signal and Sampling frequency

```
//clear//
//Example7.4:Finding the period of the sampled signal
//and Sampling frequency
clear;
close;
clc;
Wm = 2*%pi/9;
N = floor(2*%pi/(2*Wm))
disp(N,'Period of the discrete signal')
Ws = 2*%pi/N;
disp(Ws,'The Sampling frequency corresponding to the period N')
```

Scilab code Exa 7.4

Period of the sampled signal and Sampling frequency

```
//clear//
//Example7.4:Finding the period of the sampled signal
//and Sampling frequency
clear;
close;
clc;
Wm = 2*%pi/9;
N = floor(2*%pi/(2*Wm))
disp(N,'Period of the discrete signal')
Ws = 2*%pi/N;
disp(Ws,'The Sampling frequency corresponding to the period N')
```

Scilab code Exa 7.5 Multirate Signal Processing

```
//clear//
//Example7.5:Multirate Signal Processing:Sampling
   Rate Conversion
//(1)Downsampling by 4
//(2)Upsampling by 2
//(3)Upsampling by 2 and followed by downsampling by
    9
clear;
close;
clc;
Wm = 2*%pi/9;//Maximum frequency of signal
Ws = 2*Wm;   //Sampling frequency
N = floor(2*%pi/Ws);//period of discrete signal
//Original discrete time signal generation and
   Magnitude response
n = 0:0.01:N;
x = sin(Wm*n);
Wmax = 2*%pi/9;
K = 4;
k = 0:(K/1000):K;
W = k*Wmax/K;
XW = x* exp(-sqrt(-1)*n'*W);
XW_Mag = real(XW);
XW_Mag = XW_Mag/max(XW_Mag);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -
   Wmax to Wmax
XW_Mag = [mtlb_fliplr(XW_Mag), XW_Mag(2:1001)];
//(1)downsampling by 4 and corresponding magnitude
   response
n1 = 0:0.01:N/4;
y = x(1:4:length(x));
k1 = 0:(K/2000):K;
W1 = k1*4*Wmax/K;
XW4 = y* exp(-sqrt(-1)*n1'*W1);
XW4_Mag = real(XW4);
XW4_Mag = XW4_Mag/max(XW4_Mag);
W1 = [-mtlb_fliplr(W1), W1(2:$)]; // Omega from -
   Wmax to Wmax
```

```
XW4_Mag = [mtlb_fliplr(XW4_Mag), XW4_Mag(2:$)];
//(2)Upsampling by 2 and corresponding magnitude
    response
n2 = 0:0.01:2*N;
z = zeros(1,length(n2));
z([1:2:length(z)]) = x;
k2 = 0:(K/500):K;
W2 = k2*Wmax/(2*K);
XW2 = z* exp(-sqrt(-1)*n2'*W2);
XW2_Mag = real(XW2);
XW2_Mag = XW2_Mag/max(XW2_Mag);
W2 = [-mtlb_fliplr(W2), W2(2:$)]; // Omega from -
    Wmax to Wmax
XW2_Mag = [mtlb_fliplr(XW2_Mag), XW2_Mag(2:$)];
//(3)Upsampling by 2 and Downsampling by 9
    corresponding magnitude response
n3 = 0:0.01:2*N/9;
g = z([1:9:length(z)]);
k3 = 0:K/(9*500):K;
W3 = k3*9*Wmax/(2*K);
XW3 = g* exp(-sqrt(-1)*n3'*W3);
XW3_Mag = real(XW3);
XW3_Mag = XW3_Mag/max(XW3_Mag);
W3 = [-mtlb_fliplr(W3), W3(2:$)]; // Omega from -
    Wmax to Wmax
XW3_Mag = [mtlb_fliplr(XW3_Mag), XW3_Mag(2:$)];
//
figure
subplot(2,2,1)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
a.data_bounds =[-%pi,0;%pi,1.5];
plot2d(W,XW_Mag,5);
title('Spectrum of Discrete Signal X(exp(jW))')
subplot(2,2,2)
a = gca();
a.y_location ="origin";
```

```
a.x_location ="origin";
a.data_bounds =[-%pi,0;%pi,1.5];
plot2d(W1,XW4_Mag,5);
title('Spectrum of downsampled signal by 4 X(exp(jW
   /4))')
subplot(2,2,3)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
a.data_bounds =[-%pi,0;%pi,1.5];
plot2d(W2,XW2_Mag,5);
title('Spectrum of Upsampled signal by 2  X(exp(2jW)
   )')
subplot(2,2,4)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
a.data_bounds =[-%pi,0;%pi,1.5];
plot2d(W3,XW3_Mag,5);
title('Spectrum of Upsampled by 2 and Downsampled by
    9  X(exp(2jW/9))')
```

Scilab code Exa 7.5

Multirate Signal Processing

```
//clear//
//Example7.5:Multirate Signal Processing:Sampling
   Rate Conversion
//(1)Downsampling by 4
//(2)Upsampling by 2
//(3)Upsampling by 2 and followed by downsampling by
    9
clear;
close;
clc;
Wm = 2*%pi/9;//Maximum frequency of signal
Ws = 2*Wm;    //Sampling frequency
```

```
N = floor(2*%pi/Ws);//period of discrete signal
//Original discrete time signal generation and
   Magnitude response
n = 0:0.01:N;
x = sin(Wm*n);
Wmax = 2*%pi/9;
K = 4;
k = 0:(K/1000):K;
W = k*Wmax/K;
XW = x* exp(-sqrt(-1)*n'*W);
XW_Mag = real(XW);
XW_Mag = XW_Mag/max(XW_Mag);
W = [-mtlb_fliplr(W), W(2:1001)]; // Omega from -
   Wmax to Wmax
XW_Mag = [mtlb_fliplr(XW_Mag), XW_Mag(2:1001)];
//(1)downsampling by 4 and corresponding magnitude
   response
n1 = 0:0.01:N/4;
y = x(1:4:length(x));
k1 = 0:(K/2000):K;
W1 = k1*4*Wmax/K;
XW4 = y* exp(-sqrt(-1)*n1'*W1);
XW4_Mag = real(XW4);
XW4_Mag = XW4_Mag/max(XW4_Mag);
W1 = [-mtlb_fliplr(W1), W1(2:$)]; // Omega from -
   Wmax to Wmax
XW4_Mag = [mtlb_fliplr(XW4_Mag), XW4_Mag(2:$)];
//(2)Upsampling by 2 and corresponding magnitude
   response
n2 = 0:0.01:2*N;
z = zeros(1,length(n2));
z([1:2:length(z)]) = x;
k2 = 0:(K/500):K;
W2 = k2*Wmax/(2*K);
XW2 = z* exp(-sqrt(-1)*n2'*W2);
XW2_Mag = real(XW2);
XW2_Mag = XW2_Mag/max(XW2_Mag);
W2 = [-mtlb_fliplr(W2), W2(2:$)]; // Omega from -
```

```
    Wmax to Wmax
XW2_Mag = [mtlb_fliplr(XW2_Mag), XW2_Mag(2:$)];
//(3)Upsampling by 2 and Downsampling by 9
    corresponding magnitude response
n3 = 0:0.01:2*N/9;
g = z([1:9:length(z)]);
k3 = 0:K/(9*500):K;
W3 = k3*9*Wmax/(2*K);
XW3 = g* exp(-sqrt(-1)*n3'*W3);
XW3_Mag = real(XW3);
XW3_Mag = XW3_Mag/max(XW3_Mag);
W3 = [-mtlb_fliplr(W3), W3(2:$)]; // Omega from -
    Wmax to Wmax
XW3_Mag = [mtlb_fliplr(XW3_Mag), XW3_Mag(2:$)];
//
figure
subplot(2,2,1)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
a.data_bounds =[-%pi,0;%pi,1.5];
plot2d(W,XW_Mag,5);
title('Spectrum of Discrete Signal X(exp(jW))')
subplot(2,2,2)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
a.data_bounds =[-%pi,0;%pi,1.5];
plot2d(W1,XW4_Mag,5);
title('Spectrum of downsampled signal by 4 X(exp(jW
    /4))')
subplot(2,2,3)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
a.data_bounds =[-%pi,0;%pi,1.5];
plot2d(W2,XW2_Mag,5);
title('Spectrum of Upsampled signal by 2  X(exp(2jW)
    )')
subplot(2,2,4)
a = gca();
a.y_location ="origin";
a.x_location ="origin";
a.data_bounds =[-%pi,0;%pi,1.5];
plot2d(W3,XW3_Mag,5);
title('Spectrum of Upsampled by 2 and Downsampled by
     9  X(exp(2jW/9))')
```

Chapter 9

The Laplace Transform

Chapter 9

The Laplace Transform

Scilab code Exa 9.1 Lapalce Transform x(t)

```
//clear//
//Example9.1:Lapalce Transform x(t) = exp(-at).u(t)
syms t s;
a = 3;
y =laplace('%e^(-a*t)',t,s);
disp(y)
//Result
//1/(s+a)
```

Scilab code Exa 9.1

Lapalce Transform x(t)

```
//clear//
//Example9.1:Lapalce Transform x(t) = exp(-at).u(t)
syms t s;
a = 3;
y =laplace('%e^(-a*t)',t,s);
disp(y)
```

```
//Result
//1/(s+a)
```

Scilab code Exa 9.2 Lapalce Transform x(t)

```
//clear//
//Example9.2:Lapalce Transform x(t) = -exp(-at).u(-t)
syms t s;
a =3;
y = laplace('%e^(a*-t)',t,s);
disp(y)
//Result
//1/(s+a)
```

Scilab code Exa 9.2

Lapalce Transform x(t)

```
//clear//
//Example9.2:Lapalce Transform x(t) = -exp(-at).u(-t)
syms t s;
a =3;
y = laplace('%e^(a*-t)',t,s);
disp(y)
//Result
//1/(s+a)
```

Scilab code Exa 9.3 Lapalce Transform x(t)

```
//clear//
//Example9.3:Lapalce Transform x(t) = 3exp(-2t)u(t)
   -2exp(-t)u(t)
syms t s;
y = laplace('3*%e^(-2*t)-2*%e^(-t)',t,s);
disp(y)
//Result
//(3/(s+2))-(2/(s+1))
```

Scilab code Exa 9.3 Lapalce Transform x(t)

```
//clear//
//Example9.3:Lapalce Transform x(t) = 3exp(-2t)u(t)
   -2exp(-t)u(t)
syms t s;
y = laplace('3*%e^(-2*t)-2*%e^(-t)',t,s);
disp(y)
//Result
//(3/(s+2))-(2/(s+1))
```

Scilab code Exa 9.4 clear

```
//clear//
//Example9.4:Lapalce Transform x(t) = exp(-2t)u(t)+
   exp(-t)(cos3t)u(t)
syms t s;
y = laplace('%e^(-2*t)+%e^(-t)*cos(3*t)',t,s);
disp(y)
//Result
//[(s+1)/(s^2+2*s+10)]+[1/(s+2)]  refer equation
   9.29
//Equivalent to (2*s^2+5*s+12)/((s^2+2*s+10)*(s+2))
   refer equation 9.30
```

Scilab code Exa 9.4

clear

```
//clear//
//Example9.4:Lapalce Transform x(t) = exp(-2t)u(t)+
   exp(-t)(cos3t)u(t)
syms t s;
y = laplace('%e^(-2*t)+%e^(-t)*cos(3*t)',t,s);
disp(y)
//Result
//[(s+1)/(s^2+2*s+10)]+[1/(s+2)]  refer equation
   9.29
//Equivalent to (2*s^2+5*s+12)/((s^2+2*s+10)*(s+2))
   refer equation 9.30
```

Scilab code Exa 9.5 clear

```
//clear//
//Example9.5:Lapalce Transform of x(t)=s(t)-(4/3)exp
   (-t)u(t)+(1/3)exp(2t)u(t)
syms t s;
y = laplace('-(4/3)*%e^(-t)+(1/3)*%e^(2*t)',t,s);
y = 1+y;
disp(y)
//Result
//[-4/(3*(s+1))]+[1/(3*(s-2))]+1
```

Scilab code Exa 9.5

clear

```
//clear//
```

```
//Example9.5:Lapalce Transform of x(t)=s(t)-(4/3)exp
    (-t)u(t)+(1/3)exp(2t)u(t)
syms t s;
y = laplace('-(4/3)*%e^(-t)+(1/3)*%e^(2*t)',t,s);
y = 1+y;
disp(y)
//Result
//[-4/(3*(s+1))]+[1/(3*(s-2))]+1
```

Scilab code Exa 9.6 clear

```
//clear//
//Example9.6
//Lapalce Transform x(t) = exp(-at)u(t), 0<t<T
syms t s;
a = 3;
T = 10;
//t = T;
y = laplace('%e^(-a*t)-%e^(-a*t)*%e^(-(s+a)*T)',t,s)
    ;
disp(y)
//Result
// [1/(s+a)]-[(exp((-s-a)*T))/(s+a)]
```

Scilab code Exa 9.6

clear

```
//clear//
//Example9.6
//Lapalce Transform x(t) = exp(-at)u(t), 0<t<T
syms t s;
a = 3;
T = 10;
```

```
//t = T;
y = laplace('%e^(-a*t)-%e^(-a*t)*%e^(-(s+a)*T)',t,s);
disp(y)
//Result
// [1/(s+a)]-[(exp((-s-a)*T))/(s+a)]
```

Scilab code Exa 9.7 clear

```
//clear//
//Example9.7
//Lapalce Transform x(t) = exp(-b.abs(t)).u(t), 0<t<T
//x(t) = exp(-bt).u(t)+exp(bt).u(-t)
syms t s;
b = 3;
y = laplace('%e^(-b*t)-%e^(b*t)',t,s);
disp(y)
//Result
// [1/(s+b)]-[1/(s-b)]
```

Scilab code Exa 9.7

clear

```
//clear//
//Example9.7
//Lapalce Transform x(t) = exp(-b.abs(t)).u(t), 0<t<T
//x(t) = exp(-bt).u(t)+exp(bt).u(-t)
syms t s;
b = 3;
y = laplace('%e^(-b*t)-%e^(b*t)',t,s);
disp(y)
```

```
//Result
// [1/(s+b)]-[1/(s-b)]
```

Scilab code Exa 9.8 clear

```
//clear//
//Example9.8:Inverse Lapalce Transform
//X(S) = 1/((s+1)(s+2))
s =%s ;
G =syslin('c',(1/((s+1)*(s+2)))) ;
disp(G,"G( s )=")
plzr(G)
x=denom(G) ;
disp(x,"Ch a r a c t e r i s t i c s Polynomial=" )
y = roots(x) ;
disp(y," Poles of a system=" )
//Result
// -1 and -2
```

Scilab code Exa 9.8

clear

```
//clear//
//Example9.8:Inverse Lapalce Transform
//X(S) = 1/((s+1)(s+2))
s =%s ;
G =syslin('c',(1/((s+1)*(s+2)))) ;
disp(G,"G( s )=")
plzr(G)
x=denom(G) ;
disp(x,"Ch a r a c t e r i s t i c s Polynomial=" )
y = roots(x) ;
disp(y," Poles of a system=" )
```

```
//Result
// −1 and −2
```

Scilab code Exa 9.9 clear

Scilab code Exa 9.9

```
//clear//
//Example9.9:Inverse Lapalce Transform
//X(S) = 1/((s+1)(s+2))
s =%s ;
syms t ;
[A]=pfss(1/((s+1)*(s+2))) //partial fraction of F(s)
F1 = ilaplace(A(1),s,t)
F2 = ilaplace(A(2),s,t)
F=F1+F2;
disp(F,"f(t)=")
//Result
// (%eˆ−t)−(%eˆ−(2*t))
```

clear

```
//clear//
//Example9.9:Inverse Lapalce Transform
//X(S) = 1/((s+1)(s+2))
s =%s ;
syms t ;
[A]=pfss(1/((s+1)*(s+2))) //partial fraction of F(s)
F1 = ilaplace(A(1),s,t)
F2 = ilaplace(A(2),s,t)
F=F1+F2;
disp(F,"f(t)=")
//Result
// (%eˆ−t)−(%eˆ−(2*t))
```

Scilab code Exa 9.10 Inverse Lapalce Transform

```
//clear//
//Example9.10:Inverse Lapalce Transform
//X(S) = 1/((s+1)(s+2))  Re(s)< -1,Re(s)< -2
s =%s ;
syms t ;
[A]=pfss(1/((s+1)*(s+2))) //partial fraction of F(s)
F1 = ilaplace(A(1),s,t)
F2 = ilaplace(A(2),s,t)
F = -F1-F2;
disp(F,"f(t)=")
//Result
// %e^-(2*t)-%e^-t
```

Scilab code Exa 9.10

Inverse Lapalce Transform

```
//clear//
//Example9.10:Inverse Lapalce Transform
//X(S) = 1/((s+1)(s+2))  Re(s)< -1,Re(s)< -2
s =%s ;
syms t ;
[A]=pfss(1/((s+1)*(s+2))) //partial fraction of F(s)
F1 = ilaplace(A(1),s,t)
F2 = ilaplace(A(2),s,t)
F = -F1-F2;
disp(F,"f(t)=")
//Result
// %e^-(2*t)-%e^-t
```

Scilab code Exa 9.11 Inverse Lapalce Transform

```
//clear//
//Example9.11:Inverse Lapalce Transform
//X(S) = 1/((s+1)(s+2))  −2< Re(s)< −1
s =%s ;
syms t ;
[A]=pfss(1/((s+1)*(s+2))) //partial fraction of F(s)
F1 = ilaplace(A(1),s,t)
F2 = ilaplace(A(2),s,t)
F = -F1+F2;
disp(F,"f(t)=")
//Result
// −(%eˆ−t)−(%eˆ−(2*t))
```

Scilab code Exa 9.11

Inverse Lapalce Transform

```
//clear//
//Example9.11:Inverse Lapalce Transform
//X(S) = 1/((s+1)(s+2))  −2< Re(s)< −1
s =%s ;
syms t ;
[A]=pfss(1/((s+1)*(s+2))) //partial fraction of F(s)
F1 = ilaplace(A(1),s,t)
F2 = ilaplace(A(2),s,t)
F = -F1+F2;
disp(F,"f(t)=")
//Result
// −(%eˆ−t)−(%eˆ−(2*t))
```

Scilab code Exa 9.12 Inverse Lapalce Transform

Scilab code Exa 9.112

```
//clear//
//Example9.12:Inverse Lapalce Transform
//X(S) = 1/(s+(1/2))    Re(s)> -1/2
s =%s ;
G =syslin('c',(1/(s+0.5)))
disp(G,"G( s )=")
plzr(G)
```

Inverse Lapalce Transform

```
//clear//
//Example9.12:Inverse Lapalce Transform
//X(S) = 1/(s+(1/2))    Re(s)> -1/2
s =%s ;
G =syslin('c',(1/(s+0.5)))
disp(G,"G( s )=")
plzr(G)
```

Scilab code Exa 9.13 Inverse Lapalce Transform

```
//clear//
//Example9.13
//Inverse Lapalce Transform
//X1(S) = 1/(s+1)    Re(s)> -1
//X2(S) = 1/((s+1)(s+2)) Re(s)>-1
```

```
s =%s ;
syms t ;
G1 =syslin('c',(1/(s+1)));
disp(G1,"G( s )=")
figure
plzr(G1)
G2 =syslin('c',(1/((s+1)*(s+2))));
disp(G2,"G( s )=")
figure
plzr(G2)
G3 = syslin('c',(1/(s+1))-(1/((s+1)*(s+2))));
disp(G3,"G( s )=")
figure
plzr(G3)
```

Scilab code Exa 9.13

Inverse Lapalce Transform

```
//clear//
//Example9.13
//Inverse Lapalce Transform
//X1(S) = 1/(s+1)    Re(s)> -1
//X2(S) = 1/((s+1)(s+2)) Re(s)>-1
s =%s ;
syms t ;
G1 =syslin('c',(1/(s+1)));
disp(G1,"G( s )=")
figure
plzr(G1)
G2 =syslin('c',(1/((s+1)*(s+2))));
disp(G2,"G( s )=")
figure
plzr(G2)
G3 = syslin('c',(1/(s+1))-(1/((s+1)*(s+2))));
disp(G3,"G( s )=")
figure
plzr(G3)
```

Scilab code Exa 9.14 Lapalce Transform

```
//clear//
//Example9.14:Lapalce Transform
//x(t) = t.exp(-at), t>0
//x(t) = (t^2)/2.exp(-at), t>0
s =%s ;
syms t ;
a =10;
x1 = laplace('t*%e^(-10*t)',t,s);
disp(x1)
x2 = laplace('((t^2)/2)*%e^(-10*t)',t,s);
disp(x2)
//Result
//1/((s+10)^2)
// 1/((s+10)^3)
```

Scilab code Exa 9.14

Lapalce Transform

```
//clear//
//Example9.14:Lapalce Transform
//x(t) = t.exp(-at), t>0
//x(t) = (t^2)/2.exp(-at), t>0
s =%s ;
syms t ;
a =10;
x1 = laplace('t*%e^(-10*t)',t,s);
disp(x1)
x2 = laplace('((t^2)/2)*%e^(-10*t)',t,s);
disp(x2)
```

```
//Result
//1/((s+10)^2)
//  1/((s+10)^3)
```

Scilab code Exa 9.15 Inverse Lapalce Transform

Scilab code Exa 9.15

```
//clear//
//Example9.15:Inverse Lapalce Transform
//X(S) = (2s^2+5s+5)/((s+1)^2)(s+2))  Re(s)>-1
s =%s ;
syms t ;
[A]=pfss((2*(s^2)+5*s+5)/(((s+1)^2)*(s+2))); //
   partial fraction of F(s)
F1 = ilaplace(A(1),s,t)
F2 = ilaplace(A(2),s,t)
//F3 = ilaplace(A(3),s,t)
F = F1+F2;
disp(F,"f(t)=")
//Result
//(2*t*(%e^-t))-(%e^-t)+(3*%e^-(2*t))
```

Inverse Lapalce Transform

```
//clear//
//Example9.15:Inverse Lapalce Transform
//X(S) = (2s^2+5s+5)/((s+1)^2)(s+2))  Re(s)>-1
s =%s ;
syms t ;
[A]=pfss((2*(s^2)+5*s+5)/(((s+1)^2)*(s+2))); //
   partial fraction of F(s)
F1 = ilaplace(A(1),s,t)
F2 = ilaplace(A(2),s,t)
//F3 = ilaplace(A(3),s,t)
```

```
F = F1+F2;
disp(F,"f(t)=")
//Result
//(2*t*(%e^-t))-(%e^-t)+(3*%e^-(2*t))
```

Scilab code Exa 9.16 Initial Value Theorem of Lapalace Transform

Scilab code Exa 9.16
```
//clear//
//Example9.16:Initial Value Theorem of Lapalace Transform
syms s;
num =poly([12 5 2],'s','coeff')
den =poly([20 14 4 1],'s','coeff')
X = num/den
disp (X,"X(s)=")
SX = s*X;
Initial_Value =limit(SX,s,%inf);
disp(Initial_Value,"x(0)=")
//Result
//(2*%inf^3+5*%inf^2+12*%inf)/(%inf^3+4*%inf^2+14*%inf+20) =2
```

Initial Value Theorem of Lapalace Transform

```
//clear//
//Example9.16:Initial Value Theorem of Lapalace Transform
syms s;
num =poly([12 5 2],'s','coeff')
den =poly([20 14 4 1],'s','coeff')
X = num/den
disp (X,"X(s)=")
SX = s*X;
```

```
Initial_Value =limit(SX,s,%inf);
disp(Initial_Value,"x(0)=")
//Result
//(2*%inf^3+5*%inf^2+12*%inf)/(%inf^3+4*%inf^2+14*
    %inf+20) =2
```

Scilab code Exa 9.17 Analysis and Characterization of LTI System

```
//clear//
//Example9.17:Analysis and Characterization of LTI
    System
//Lapalce Transform h(t) = exp(-t).u(t)
syms t s;
h =laplace('%e^(-t)',t,s);
disp(h)
//Result
//1/(s+1)
```

Scilab code Exa 9.17

Analysis and Characterization of LTI System

```
//clear//
//Example9.17:Analysis and Characterization of LTI
    System
//Lapalce Transform h(t) = exp(-t).u(t)
syms t s;
h =laplace('%e^(-t)',t,s);
disp(h)
//Result
//1/(s+1)
```

Scilab code Exa 9.18 Analysis and Characterization of LTI System

```
//clear//
//Example9.18:Analysis and Characterization of LTI
   System
//Lapalce Transform x(t) = exp(-abs(t))
//x(t) = exp(-t).u(t)+exp(t).u(-t)
syms t s;
y = laplace('%e^(-t)-%e^(t)',t,s);
disp(y)
//Result
//   (1/(s+1))-(1/(s-1))
```

Scilab code Exa 9.18

Analysis and Characterization of LTI System

```
//clear//
//Example9.18:Analysis and Characterization of LTI
   System
//Lapalce Transform x(t) = exp(-abs(t))
//x(t) = exp(-t).u(t)+exp(t).u(-t)
syms t s;
y = laplace('%e^(-t)-%e^(t)',t,s);
disp(y)
//Result
//   (1/(s+1))-(1/(s-1))
```

Scilab code Exa 9.19 Analysis and Characterization of LTI System

```
//clear//
```

```
//Example9.19: Analysis and Characterization of LTI
    System
//Inverse Lapalce Transform
//X(S) = (e^s)/(s+1)
syms s t ;
h1 = exp(-1);//Inverse Laplace Transform of exp(s)
H2 =1/(s+1);
h2 = ilaplace(H2,s,t)
h = h1*h2;
disp(h,"h(t)=")
//Result
// (18089*(%e^-t))/49171 =  0.3678794(%e^-t)
```

Scilab code Exa 9.19

Analysis and Characterization of LTI System

```
//clear//
//Example9.19: Analysis and Characterization of LTI
    System
//Inverse Lapalce Transform
//X(S) = (e^s)/(s+1)
syms s t ;
h1 = exp(-1);//Inverse Laplace Transform of exp(s)
H2 =1/(s+1);
h2 = ilaplace(H2,s,t)
h = h1*h2;
disp(h,"h(t)=")
//Result
// (18089*(%e^-t))/49171 =  0.3678794(%e^-t)
```

Scilab code Exa 9.20 **Scilab code Exa 9.20** Inverse Lapalce Transform
Inverse Lapalce Transform

```
//clear//
//Example9.20:Inverse Lapalce Transform
//X(S) = ((s-1)/((s+1)*(s-2)))
s =%s ;
syms t ;
[A] = pfss(s/((s+1)*(s-2)));
[B] = pfss(1/((s+1)*(s-2)));
F1 = ilaplace(A(1),s,t)
F2 = ilaplace(A(2),s,t)
F3 = ilaplace(B(1),s,t)
F4 = ilaplace(B(2),s,t)
F = F1+F2-F3-F4;
disp(F,"f(t)=")
//Result
//f(t)= 33333329999999*exp(2*t)/99999970000000+6666664*%e^-t/9999997
//i.e. f(t) =0.3333334*exp(2*t)+0.6666666*%e^(-t)
//Refer equation 9.120. (1/3)=0.3333 and (2/3) = 0.66666
```

```
//clear//
//Example9.20:Inverse Lapalce Transform
//X(S) = ((s-1)/((s+1)*(s-2)))
s =%s ;
syms t ;
[A] = pfss(s/((s+1)*(s-2)));
[B] = pfss(1/((s+1)*(s-2)));
F1 = ilaplace(A(1),s,t)
F2 = ilaplace(A(2),s,t)
F3 = ilaplace(B(1),s,t)
F4 = ilaplace(B(2),s,t)
F = F1+F2-F3-F4;
disp(F,"f(t)=")
//Result
//f(t)= 33333329999999*exp(2*t)/99999970000000+6666664*%e^-t/9999997
//i.e. f(t) =0.3333334*exp(2*t)+0.6666666*%e^(-t)
```

```
//Refer equation 9.120. (1/3)=0.3333 and (2/3) =
    0.66666
```

Scilab code Exa 9.21 Analysis and Characterization of LTI System

```
//clear//
//Example9.21:Analysis and Characterization of LTI
    System
//Lapalce Transform h(t) = exp(2t)u(t), Re(s)>2
syms t s;
X = laplace('%e^(2*t)',t,s);
disp(X)
//Result
//1/(s-2)
```

Scilab code Exa 9.21

Analysis and Characterization of LTI System

```
//clear//
//Example9.21:Analysis and Characterization of LTI
    System
//Lapalce Transform h(t) = exp(2t)u(t), Re(s)>2
syms t s;
X = laplace('%e^(2*t)',t,s);
disp(X)
//Result
//1/(s-2)
```

Scilab code Exa 9.25 Finding Transfer function H(S) of LTI system

```
//clear//
//Example9.25:LTI Systems Characterized by Linear
    Constant
//Coefficient differential Equation
//Finding Transfer function H(S) of LTI system
//x(t) = exp(-3t).u(t)
//y(t) = [exp(-t)-exp(-2t)].u(t)
syms t s;
X = laplace('%e^(-3*t)',t,s);
Y = laplace('%e^(-t)-%e^(-2*t)',t,s);
H = Y/X;
disp(H)
//Result
//(s+3)*(1/(s+1)-1/(s+2))
```

Scilab code Exa 9.25

Finding Transfer function H(S) of LTI system

```
//clear//
//Example9.25:LTI Systems Characterized by Linear
    Constant
//Coefficient differential Equation
//Finding Transfer function H(S) of LTI system
//x(t) = exp(-3t).u(t)
//y(t) = [exp(-t)-exp(-2t)].u(t)
syms t s;
X = laplace('%e^(-3*t)',t,s);
Y = laplace('%e^(-t)-%e^(-2*t)',t,s);
H = Y/X;
disp(H)
//Result
//(s+3)*(1/(s+1)-1/(s+2))
```

Scilab code Exa 9.31 Partial Fraction

```
//clear//
//Example9.31:Causal LTI Systems described by differential equations
//and Rational System functions
//Partial Fraction
//H(S) = ((s-1)/((s+1)*(s-2)))
s =%s ;
syms t ;
[A] = pfss((2*s^2+4*s-6)/(s^2+3*s+2));
disp(A,"H(S)=")
//Result  H(S)=
//// - 8
//     -----
//     1 + s
//      6
//     -----
//     2 + s
//
//     2
```

Scilab code Exa 9.31

Partial Fraction

```
//clear//
//Example9.31:Causal LTI Systems described by differential equations
//and Rational System functions
//Partial Fraction
//H(S) = ((s-1)/((s+1)*(s-2)))
s =%s ;
syms t ;
[A] = pfss((2*s^2+4*s-6)/(s^2+3*s+2));
disp(A,"H(S)=")
//Result  H(S)=
//// - 8
```

```
//      -----
//      1 + s
//        6
//      -----
//      2 + s
//
//      2
```

Scilab code Exa 9.33 Unilateral Laplace Transform

```
//clear//
//Example9.33:Unilateral Laplace Transform:Time Shifting Property
//x(t) = exp(-a(t+1)).u(t+1)
syms t s;
a = 2;
X = laplace('%e^(-a*(t+1))',t,s);
disp(X)
//Result
//%e^-a/(s+a)
```

Scilab code Exa 9.33

Unilateral Laplace Transform

```
//clear//
//Example9.33:Unilateral Laplace Transform:Time Shifting Property
//x(t) = exp(-a(t+1)).u(t+1)
syms t s;
a = 2;
X = laplace('%e^(-a*(t+1))',t,s);
disp(X)
//Result
```

```
//%e^-a/(s+a)
```

Scilab code Exa 9.34 Unilateral Laplace Transform

```
//clear//
//Example9.34:Unilateral Laplace Transform
//x(t) = s(t)+2u(t)+e^t.u(t)
syms t s;
a = 2;
X = laplace('2+%e^(t)',t,s);
Y = 1+X;
disp(X)
disp(Y)
//Result
// (2/s)+(1/(s-1))+1
```

Scilab code Exa 9.34

Unilateral Laplace Transform

```
//clear//
//Example9.34:Unilateral Laplace Transform
//x(t) = s(t)+2u(t)+e^t.u(t)
syms t s;
a = 2;
X = laplace('2+%e^(t)',t,s);
Y = 1+X;
disp(X)
disp(Y)
//Result
// (2/s)+(1/(s-1))+1
```

Scilab code Exa 9.35 clear

```
//clear//
//Example9.35:Unilateral Inverse Laplace Transform
//X(S) = 1/((s+1)(s+2))
s = %s;
syms t;
X = 1/((s+1)*(s+2));
x = ilaplace(X,s,t);
disp(X)
disp(x)
//Result
// (%e^-t)-(%e^-(2*t))
```

Scilab code Exa 9.35

clear

```
//clear//
//Example9.35:Unilateral Inverse Laplace Transform
//X(S) = 1/((s+1)(s+2))
s = %s;
syms t;
X = 1/((s+1)*(s+2));
x = ilaplace(X,s,t);
disp(X)
disp(x)
//Result
// (%e^-t)-(%e^-(2*t))
```

Scilab code Exa 9.36 clear

```
//clear//
//Example9.36:Unilateral  Laplace Transform
//X(S) = ((s^2)-3)/(s+2)
s = %s;
syms t;
[X] = pfss(((s^2)-3)/(s+2));
disp(X)
```

Scilab code Exa 9.36

clear

```
//clear//
//Example9.36:Unilateral  Laplace Transform
//X(S) = ((s^2)-3)/(s+2)
s = %s;
syms t;
[X] = pfss(((s^2)-3)/(s+2));
disp(X)
```

Scilab code Exa 9.37 clear

```
//clear//
//Example9.37:Unilateral  Laplace Transform:Solving Differential Equation
//Y(S) = alpha/(s(s+1)(s+2))
s = %s;
syms t;
alpha = 1;   //Alpha value assigned as some constant one
[A] = pfss(alpha/(s*(s+1)*(s+2)));
F1 = ilaplace(A(1),s,t)
F2 = ilaplace(A(2),s,t)
F3 = ilaplace(A(3),s,t)
```

```
F = F1+F2+F3
disp(F)
//result
// (-%e^-t)+((%e^-(2*t))/2)+(1/2 )
```

Scilab code Exa 9.37

clear

```
//clear//
//Example9.37:Unilateral  Laplace Transform:Solving
    Differential Equation
//Y(S) = alpha/(s(s+1)(s+2))
s = %s;
syms t;
alpha = 1;   //Alpha value assigned as some constant
     one
[A] = pfss(alpha/(s*(s+1)*(s+2)));
F1 = ilaplace(A(1),s,t)
F2 = ilaplace(A(2),s,t)
F3 = ilaplace(A(3),s,t)
F = F1+F2+F3
disp(F)
//result
// (-%e^-t)+((%e^-(2*t))/2)+(1/2 )
```

Scilab code Exa 9.38 clear

```
//clear//
//Example9.38:Unilateral  Laplace Transform:Solving
    Differential Equation
//Y(S)=[beta(s+3)/((s+1)(s+2))]+[gamma/((s+2)(s+2))
    ]+[alpha/(s(s+1)(s+2))]
s = %s;
```

```
syms t;
alpha = 2; //input constant
beta_B = 3; //intial condition
gamma_v = -5; //initial condition
Y1 = 1/s;
Y2 = 1/(s+1);
Y3 = 3/(s+2);
Y = Y1-Y2+Y3;
disp(Y)
y = ilaplace(Y,s,t)
disp(y)
//result
// ( -%e^(-t))+3*(%e^(-(2*t)))+1
```

Scilab code Exa 9.38

clear

```
//clear//
//Example9.38:Unilateral  Laplace Transform:Solving
   Differential Equation
//Y(S)=[beta(s+3)/((s+1)(s+2))]+[gamma/((s+2)(s+2))
   ]+[alpha/(s(s+1)(s+2))]
s = %s;
syms t;
alpha = 2; //input constant
beta_B = 3; //intial condition
gamma_v = -5; //initial condition
Y1 = 1/s;
Y2 = 1/(s+1);
Y3 = 3/(s+2);
Y = Y1-Y2+Y3;
disp(Y)
y = ilaplace(Y,s,t)
disp(y)
//result
// ( -%e^(-t))+3*(%e^(-(2*t)))+1
```

Chapter 10

The Z Transform

Chapter 11

The Z Transform

Scilab code Exa 10.1 Scilab code Exa 10.1 Ztransform of x[n]

```
//clear//
// Example10.1:Ztransform of x[n]  = (a)^n.u[n]
syms n z;
a = 0.5;
x =(a)^n
X = symsum(x*(z^(-n)),n,0,%inf)
disp(X,"ans=")
//Result
//1.0*(2^(-%inf-1)*z^(-%inf-1)-1)/(1/(2*z)-1)
//Equivalent to -1/(0.5*(z^-1)-1)
```

Ztransform of x[n]

```
//clear//
// Example10.1:Ztransform of x[n]  = (a)^n.u[n]
syms n z;
a = 0.5;
x =(a)^n
X = symsum(x*(z^(-n)),n,0,%inf)
disp(X,"ans=")
```

```
//Result
//1.0*(2^(-%inf-1)*z^(-%inf-1)-1)/(1/(2*z)-1)
//Equivalent to -1/(0.5*(z^-1)-1)
```

Scilab code Exa 10.2 Scilab code Exa 10.2 Z transform of x[n] = -a$^{n}.u[-n-1]$

```
//clear//
//Example 10.2:Z transform of x[n] = -a^n. u[-n-1]
//a = 0.5
clear;
close;
clc;
syms n z;
a = 0.5;
x=-(0.5)^(-n)
X=symsum(x*(z^(n)),n,1,%inf)
disp(X,"ans=")
//Result
// -1.0*(2^(%inf+1)*z^(%inf+1)-2*z)/(2*z-1)
//Equivalent to -1*-2*z/(2*z-1) = 1/(1-0.5*z^-1)
```

Scilab code Exa 10.3 Scilab code Exa 10.3 Z transform of x[n]

```
//clear//
//Example 10.3:Z transform of x[n] = 7.(1/3)^n.u[n
    ]-6.(1/2)^n.u[n]
syms n z;
```

```
x1=(0.33)^(n)
X1=symsum(7*x1*(z^(-n)),n,0,%inf)
x2=(0.5)^(n)
X2=symsum(6*x2*(z^(-n)),n,0,%inf)
X = X1-X2
disp(X,"ans=")
//Result
// -6.0*(2^(-%inf-1)*z^(-%inf-1)-1)/(1/(2*z)-1)
//Equivalent to -6*-1/(0.5*z^-1 -1)
//The Region of Convergence is |z|>1/2
```

Z transform of x[n]

```
//clear//
//Example 10.3:Z transform of x[n] = 7.(1/3)^n.u[n
   ]-6.(1/2)^n.u[n]
syms n z;
x1=(0.33)^(n)
X1=symsum(7*x1*(z^(-n)),n,0,%inf)
x2=(0.5)^(n)
X2=symsum(6*x2*(z^(-n)),n,0,%inf)
X = X1-X2
disp(X,"ans=")
//Result
// -6.0*(2^(-%inf-1)*z^(-%inf-1)-1)/(1/(2*z)-1)
//Equivalent to -6*-1/(0.5*z^-1 -1)
//The Region of Convergence is |z|>1/2
```

Scilab code Exa 10.4 Z-transform of sine signal

```
//clear//
//Example10.4:Z-transform of sine signal
syms n z;
Wo =%pi/4;
```

```
a = (0.33)^n;
x1=%e^(sqrt(-1)*Wo*n);
X1=symsum(a*x1*(z^(-n)),n,0,%inf)
x2=%e^(-sqrt(-1)*Wo*n)
X2=symsum(a*x2*(z^(-n)),n,0,%inf)
X =(1/(2*sqrt(-1)))*(X1-X2)
disp(X,"ans=")
```

Scilab code Exa 10.4

Z-transform of sine signal

```
//clear//
//Example10.4:Z−transform of sine signal
syms n z;
Wo =%pi/4;
a = (0.33)^n;
x1=%e^(sqrt(-1)*Wo*n);
X1=symsum(a*x1*(z^(-n)),n,0,%inf)
x2=%e^(-sqrt(-1)*Wo*n)
X2=symsum(a*x2*(z^(-n)),n,0,%inf)
X =(1/(2*sqrt(-1)))*(X1-X2)
disp(X,"ans=")
```

Scilab code Exa 10.5 Z-transform of Impulse Sequence

```
//clear//
//Example10.5:Z−transform of Impulse Sequence
syms n z;
X=symsum(1*(z^(-n)),n,0,0);
disp(X,"ans=")
//Result
// 1
```

Scilab code Exa 10.5

Z-transform of Impulse Sequence

```
//clear//
//Example10.5:Z-transform of Impulse Sequence
syms n z;
X=symsum(1*(z^(-n)),n,0,0);
disp(X,"ans=")
//Result
// 1
```

Scilab code Exa 10.6 **Scilab code Exa 10.6** Z transform of x[n] Z transform of x[n]

```
//clear//
//Example 10.6:Z transform of x[n] = a^n, 0 < n < N
    -1
syms n z;
a = 0.5;
N =6;
x=(a)^(n)
X=symsum(x*(z^(-n)),n,0,N)
disp(X,"ans=")
//Result
//0.5/z+0.25/z^2+0.125/z^3+0.0625/z^4+0.03125/z
    ^5+0.015625/z^6+1.0
```

```
//clear//
//Example 10.6:Z transform of x[n] = a^n, 0 < n < N
    -1
syms n z;
a = 0.5;
```

```
N =6;
x=(a)^(n)
X=symsum(x*(z^(-n)),n,0,N)
disp(X,"ans=")
//Result
//0.5/z+0.25/z^2+0.125/z^3+0.0625/z^4+0.03125/z
    ^5+0.015625/z^6+1.0
```

Scilab code Exa 10.7 **Scilab code Exa 10.7** Z transform of x[n]

```
//clear//
//Example 10.7:Z transform of x[n] = b^n.u[n]+b^-n.u
    [-n-1]
syms n z;
b = 0.5;
x1=(b)^(n)
x2=(b)^(-n)
X1=symsum(x1*(z^(-n)),n,0,%inf)
X2=symsum(x2*(z^(n)),n,1,%inf)
X = X1+X2;
disp(X,"ans=")
//Result
//+1.0*(2^(-%inf-1)*z^(-%inf-1)-1)/(1/(2*z)-1)
//Equivalent to -1/(0.5*z^-1 - 1)
//Region of Convergence |z|>0.5
```

Z transform of x[n]

```
//clear//
//Example 10.7:Z transform of x[n] = b^n.u[n]+b^-n.u
    [-n-1]
syms n z;
b = 0.5;
x1=(b)^(n)
```

```
x2=(b)^(-n)
X1=symsum(x1*(z^(-n)),n,0,%inf)
X2=symsum(x2*(z^(n)),n,1,%inf)
X = X1+X2;
disp(X,"ans=")
//Result
//+1.0*(2^(-%inf-1)*z^(-%inf-1)-1)/(1/(2*z)-1)
//Equivalent to -1/(0.5*z^-1 - 1)
//Region of Convergence |z|>0.5
```

Scilab code Exa 10.9 clear

```
//clear//
//Example10.9:Inverse Z Transform:ROC |z|>1/3
z = %z;
syms n z1;//To find out Inverse z transform z must
   be linear z = z1
X  =z*(3*z-(5/6))/((z-(1/4))*(z-(1/3)))
X1 = denom(X);
zp = roots(X1);
X1 = z1*(3*z1-(5/6))/((z1-(1/4))*(z1-(1/3)))
F1 = X1*(z1^(n-1))*(z1-zp(1));
F2 = X1*(z1^(n-1))*(z1-zp(2));
h1 = limit(F1,z1,zp(1));
disp(h1,'h1[n]=')
h2 = limit(F2,z1,zp(2));
disp(h2,'h2[n]=')
h = h1+h2;
disp(h,'h[n]=')
////Result
//h[n]= (1/4)^n+(2/3)^n
```

Scilab code Exa 10.9 clear

```
//clear//
//Example10.9:Inverse Z Transform:ROC |z|>1/3
z = %z;
syms n z1;//To find out Inverse z transform z must
   be linear z = z1
X  =z*(3*z-(5/6))/((z-(1/4))*(z-(1/3)))
X1 = denom(X);
zp = roots(X1);
X1 = z1*(3*z1-(5/6))/((z1-(1/4))*(z1-(1/3)))
F1 = X1*(z1^(n-1))*(z1-zp(1));
F2 = X1*(z1^(n-1))*(z1-zp(2));
h1 = limit(F1,z1,zp(1));
disp(h1,'h1[n]=')
h2 = limit(F2,z1,zp(2));
disp(h2,'h2[n]=')
h = h1+h2;
disp(h,'h[n]=')
////Result
//h[n]= (1/4)^n+(2/3)^n
```

Scilab code Exa 10.10 Inverse Z Transform

```
//clear//
//Example10.10:Inverse Z Transform:ROC 1/4<|z|<1/3
z = %z;
syms n z1;//To find out Inverse z transform z must
   be linear z = z1
X  =z*(3*z-(5/6))/((z-(1/4))*(z-(1/3)))
X1 = denom(X);
zp = roots(X1);
X1 = z1*(3*z1-(5/6))/((z1-(1/4))*(z1-(1/3)))
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-1)','h2[n]=')
disp((h1)*'u(n)'-(h2)*'u(n-1)','h[n]=')
////Result
// h[n]=  u(n)/4^n-2*u(n-1)/3^n
//Equivalent to h[n] =(1/4)^n.u[n]-2*(1/3)^n.u[-n-1]
```

Scilab code Exa 10.10

Inverse Z Transform

```
//clear//
//Example10.10:Inverse Z Transform:ROC 1/4<|z|<1/3
z = %z;
syms n z1;//To find out Inverse z transform z must
   be linear z = z1
X  =z*(3*z-(5/6))/((z-(1/4))*(z-(1/3)))
X1 = denom(X);
zp = roots(X1);
X1 = z1*(3*z1-(5/6))/((z1-(1/4))*(z1-(1/3)))
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-1)','h2[n]=')
disp((h1)*'u(n)'-(h2)*'u(n-1)','h[n]=')
////Result
// h[n]=  u(n)/4^n-2*u(n-1)/3^n
//Equivalent to h[n] =(1/4)^n.u[n]-2*(1/3)^n.u[-n-1]
```

Scilab code Exa 10.11 Inverse Z Transform

```
//clear//
//Example10.11:Inverse Z Transform:ROC |z|<1/4
z = %z;
syms n z1;//To find out Inverse z transform z must
   be linear z = z1
X  =z*(3*z-(5/6))/((z-(1/4))*(z-(1/3)))
X1 = denom(X);
zp = roots(X1);
X1 = z1*(3*z1-(5/6))/((z1-(1/4))*(z1-(1/3)))
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-1)','h1[n]=')
h2 = limit(F2,z1,zp(2));
disp((h2)*'u(-n-1)','h2[n]=')
disp(-(h1)*'u(-n-1)'-(h2)*'u(-n-1)','h[n]=')
////Result
// h[n]=  -u(-n-1)/4^n-2*u(-n-1)/3^n
//Equivalent to h[n] =-(1/4)^n.u[-n-1]-2*(1/3)^n.u[-
   n-1]
```

Scilab code Exa 10.11 Inverse Z Transform

```
//clear//
//Example10.11:Inverse Z Transform:ROC |z|<1/4
z = %z;
syms n z1;//To find out Inverse z transform z must
   be linear z = z1
X  =z*(3*z-(5/6))/((z-(1/4))*(z-(1/3)))
X1 = denom(X);
zp = roots(X1);
X1 = z1*(3*z1-(5/6))/((z1-(1/4))*(z1-(1/3)))
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-1)','h1[n]=')
h2 = limit(F2,z1,zp(2));
disp((h2)*'u(-n-1)','h2[n]=')
disp(-(h1)*'u(-n-1)'-(h2)*'u(-n-1)','h[n]=')
////Result
// h[n]=  -u(-n-1)/4^n-2*u(-n-1)/3^n
//Equivalent to h[n] =-(1/4)^n.u[-n-1]-2*(1/3)^n.u[-
    n-1]
```

Scilab code Exa 10.12 Inverse z tranform

```
//clear//
//Example10.12:Inverse z tranform:For Finite
    duration discrete sequence
syms z;
X = [4*z^2 0 2 3*z^-1];
n = -2:1;
for i = 1:length(X)
  x(i) = X(i)*(z^n(i));
end
disp(x,'x[n]=')
```

Scilab code Exa 10.12

Inverse z tranform

```
//clear//
//Example10.12:Inverse z tranform:For Finite
    duration discrete sequence
syms z;
X = [4*z^2 0 2 3*z^-1];
n = -2:1;
for i = 1:length(X)
  x(i) = X(i)*(z^n(i));
end
disp(x,'x[n]=')
```

Scilab code Exa 10.13 Inverse z tranform of InFinite duration discrete sequence

```
//clear//
//Example10.13:Inverse z tranform ofInFinite
   duration discrete sequence
//Power Series Method (OR)//Long Division Method
z = %z;
a = 2;
X = ldiv(z,z-a,5)
```

Scilab code Exa 10.13

Inverse z tranform of InFinite duration discrete sequence

```
//clear//
//Example10.13:Inverse z tranform ofInFinite
   duration discrete sequence
//Power Series Method (OR)//Long Division Method
z = %z;
a = 2;
X = ldiv(z,z-a,5)
```

Scilab code Exa 10.18 Ztransform-Differentiation Property

```
//clear//
// Example10.18:Ztransform-Differentiation Property
// x[n]  = (a)^n.u[n]
```

```
syms n z;
a = 0.5;
x =(a)^n
X = symsum(x*(z^(-n)),n,0,%inf)
X1 = -1/((1/(2*z))-1)        //z transform of 0.5^n.u[
   n]
Y  = -z*diff(X,z)  //Differentiation property of z-
   transform
disp(X,"ans=")
disp(Y,"ans=")
//Result
//X(z) = 1.0*(2^(-%inf-1)*z^(-%inf-1)-1)/(1/(2*z)-1)
//Y(z) = -1.0*(-%inf-1)*2^(-%inf-1)*z^(-%inf-1)
   /(1/(2*z)-1)
//Y1(z) = 1/(2*(1/(2*z)-1)^2*z)
//Equivalent to Y1(z) = 0.5*z^-1/((1-0.5*z^-1)^2)
```

Scilab code Exa 10.18

Ztransform-Differentiation Property

```
//clear//
// Example10.18:Ztransform-Differentiation Property
// x[n] = (a)^n.u[n]
syms n z;
a = 0.5;
x =(a)^n
X = symsum(x*(z^(-n)),n,0,%inf)
X1 = -1/((1/(2*z))-1)        //z transform of 0.5^n.u[
   n]
Y  = -z*diff(X,z)  //Differentiation property of z-
   transform
disp(X,"ans=")
disp(Y,"ans=")
//Result
//X(z) = 1.0*(2^(-%inf-1)*z^(-%inf-1)-1)/(1/(2*z)-1)
//Y(z) = -1.0*(-%inf-1)*2^(-%inf-1)*z^(-%inf-1)
   /(1/(2*z)-1)
```

```
//Y1(z) = 1/(2*(1/(2*z)-1)^2*z)
//Equivalent to Y1(z) = 0.5*z^-1/((1-0.5*z^-1)^2)
```

Scilab code Exa 10.19 Z Transform : Initial Value Theorem

```
//clear//
//Example10.19:Z Transform : Initial Value Theorem
z = %z;
syms n z1;//To find out Inverse z transform z must
   be linear z = z1
X  =z*(z-(3/2))/((z-(1/3))*(z-(1/2)))
X1 = denom(X);
zp = roots(X1);
X1 = z1*(z1-(3/2))/((z1-(1/3))*(z1-(1/2)))
F1 = X1*(z1^(n-1))*(z1-zp(1));
F2 = X1*(z1^(n-1))*(z1-zp(2));
x1 = limit(F1,z1,zp(1));
x2 = limit(F2,z1,zp(2));
x = x1+x2;
disp(x,'x[n]=')
x_initial = limit(x,n,0);
disp(x_initial,'x[0]=')
////Result
//x[n]= 7/3^n-3*2^(1-n)
//x[0]= 1; Initial Value
```

Scilab code Exa 10.19

Z Transform : Initial Value Theorem

```
//clear//
//Example10.19:Z Transform : Initial Value Theorem
z = %z;
```

```
syms n z1;//To find out Inverse z transform z must
   be linear z = z1
X  =z*(z-(3/2))/((z-(1/3))*(z-(1/2)))
X1 = denom(X);
zp = roots(X1);
X1 = z1*(z1-(3/2))/((z1-(1/3))*(z1-(1/2)))
F1 = X1*(z1^(n-1))*(z1-zp(1));
F2 = X1*(z1^(n-1))*(z1-zp(2));
x1 = limit(F1,z1,zp(1));
x2 = limit(F2,z1,zp(2));
x = x1+x2;
disp(x,'x[n]=')
x_initial = limit(x,n,0);
disp(x_initial,'x[0]=')
////Result
//x[n]= 7/3^n-3*2^(1-n)
//x[0]= 1; Initial Value
```

Scilab code Exa 10.23 Inverse Z Transform H(z) =z/z-a

```
//clear//
//Example10.23:Inverse Z Transform H(z) =z/z-a
//z = %z;
syms n z;
a = 2;
H = z/(z-a);
F = H*z^(n-1)*(z-a);
h = limit(F,z,a);
disp(h,'h[n]=')
```

Scilab code Exa 10.23

Inverse Z Transform H(z) =z/z-a

```
//clear//
//Example10.23:Inverse Z Transform H(z) =z/z−a
//z = %z;
syms n z;
a = 2;
H = z/(z-a);
F = H*z^(n-1)*(z-a);
h = limit(F,z,a);
disp(h,'h[n]=')
```

Scilab code Exa 10.25 Coefficient Difference equations

```
//clear//
//Example10.25:LTi Systems characterized by Linear
   Constant
//Coefficient Difference equations
//Inverse Z Transform
//z = %z;
syms n z;
H1 = z/(z-(1/2));
H2 = (1/3)/(z-(1/2));
F1 = H1*z^(n-1)*(z-(1/2));
F2 = H2*z^(n-1)*(z-(1/2));
h1 = limit(F1,z,1/2);
disp(h1,'h1[n]=')
h2 = limit(F2,z,1/2);
disp(h2,'h2[n]=')
h = h1+h2;
disp(h,'h[n]=')
//Result
//h[n]=  [(1/2)^n]+[2^(1−n)]/3
//Which is Equivalent to h[n] =[(1/2)^n]+[(1/2)^(n
   −1)]/3
```

Scilab code Exa 10.25 Coefficient Difference equations

```
//clear//
//Example10.25:LTi Systems characterized by Linear
    Constant
//Coefficient Difference equations
//Inverse Z Transform
//z = %z;
syms n z;
H1 = z/(z-(1/2));
H2 = (1/3)/(z-(1/2));
F1 = H1*z^(n-1)*(z-(1/2));
F2 = H2*z^(n-1)*(z-(1/2));
h1 = limit(F1,z,1/2);
disp(h1,'h1[n]=')
h2 = limit(F2,z,1/2);
disp(h2,'h2[n]=')
h = h1+h2;
disp(h,'h[n]=')
//Result
//h[n]=  [(1/2)^n]+[2^(1-n)]/3
//Which is Equivalent to h[n] =[(1/2)^n]+[(1/2)^(n
    -1)]/3
```

Scilab code Exa 10.33 Differentiation Property of Unilateral Ztransform

```
//clear//
// Example10.33:Differentiation Property of
    Unilateral Ztransform
// x[n]  = (a)^(n+1).u[n+1]
syms n z;
a = 0.5;
```

```
x =(a)^(n+1)
X = symsum(x*(z^(-n)),n,-1,%inf)
disp(X,"ans=")
//Result
//X(z)= 0.5*(2^(-%inf-1)*z^(-%inf-1)-2*z)/(1/(2*z)
    -1)
//Equivalent to z/(1-0.5*z^-1)
```

Scilab code Exa 10.33

Differentiation Property of Unilateral Ztransform

```
//clear//
// Example10.33:Differentiation Property of
    Unilateral Ztransform
// x[n]  = (a)^(n+1).u[n+1]
syms n z;
a = 0.5;
x =(a)^(n+1)
X = symsum(x*(z^(-n)),n,-1,%inf)
disp(X,"ans=")
//Result
//X(z)= 0.5*(2^(-%inf-1)*z^(-%inf-1)-2*z)/(1/(2*z)
    -1)
//Equivalent to z/(1-0.5*z^-1)
```

Scilab code Exa 10.34 Unilateral Ztransform- partial fraction

```
//clear//
// Example10.34:Unilateral Ztransform- partial
    fraction
// X(z) =(3-(5/6)*(z^-1))/((1-(1/4)*(z^-1))*(1-(1/3)
    *(z^-1)))
z = %z;
```

```
s = %s;
syms n t;
a = 0.5;
[A]=pfss((3-(5/6)*(z^-1))/((1-(1/4)*(z^-1))*(1-(1/3)
    *(z^-1))))
x1 = horner(A(1),z)
x2 = horner(A(2),z)
x3 = A(3)
x = x1+x2+x3
disp(x1,"ans=")
disp(x2,"ans=")
disp(x3,"ans=")
disp(x,"ans=")
//Result

//       0.6666667
//     ─────────────
//   - 0.3333333 + z

//       0.25
//     ────────
//   - 0.25 + z

//3

//sum of these, gives the original value
//                          2
//        - 0.8333333z + 3z
//     ─────────────────────────
//                                 2
//      0.0833333 - 0.5833333z + z
```

Scilab code Exa 10.34

Unilateral Ztransform- partial fraction

```
//clear//
```

```
// Example10.34:Unilateral Ztransform- partial
   fraction
// X(z) =(3-(5/6)*(z^-1))/((1-(1/4)*(z^-1))*(1-(1/3)
   *(z^-1)))
z = %z;
s = %s;
syms n t;
a = 0.5;
[A]=pfss((3-(5/6)*(z^-1))/((1-(1/4)*(z^-1))*(1-(1/3)
   *(z^-1))))
x1 = horner(A(1),z)
x2 = horner(A(2),z)
x3 = A(3)
x = x1+x2+x3
disp(x1,"ans=")
disp(x2,"ans=")
disp(x3,"ans=")
disp(x,"ans=")
//Result

//      0.6666667
//    -------------
//  - 0.3333333 + z

//      0.25
//    --------
//  - 0.25 + z

//3

//sum of these, gives the original value
//                             2
//       - 0.8333333z + 3z
//    -------------------------
//                                  2
//     0.0833333 - 0.5833333z + z
```

Scilab code Exa 10.36 Output response of an LTI System

```
//clear//
//Example 10.36:To find output response of an LTI
    System
syms n z;
H = z/(z+3)
X = z/(z-1)
Y = X*H
F1 = Y*(z^(n-1))*(z-1);
y1 = limit(F1,z,1);
F2 = Y*(z^(n-1))*(z+3);
y2 = limit(F2,z,-3);
disp(y1*"u(n)"+y2*"u(n)",'y[n]=')
//Result
//y[n] = u(n)/4-(-3)^(n+1)*u(n)/4
//Equivalent to = (1/4).u[n]-(3/4)(-3)^n.u[n]
```

Scilab code Exa 10.36

Output response of an LTI System

```
//clear//
//Example 10.36:To find output response of an LTI
    System
syms n z;
H = z/(z+3)
X = z/(z-1)
Y = X*H
F1 = Y*(z^(n-1))*(z-1);
y1 = limit(F1,z,1);
F2 = Y*(z^(n-1))*(z+3);
y2 = limit(F2,z,-3);
```

```
disp(y1*"u(n)"+y2*"u(n)",'y[n]=')
//Result
//y[n] = u(n)/4-(-3)^(n+1)*u(n)/4
//Equivalent to = (1/4).u[n]-(3/4)(-3)^n.u[n]
```

Scilab code Exa 10.37 Output response of an LTI System

```
//clear//
//Example 10.37:To find output response of an LTI System
syms n z;
alpha = 8; //input constant
beta_b = 1; //initial condition y[-1] = 1
Y1 = -((3*beta_b*z)/(z+3))
Y2 = (alpha*z^2/((z+3)*(z-1)))
F1 = Y1*(z^(n-1))*(z+3);
y1 = limit(F1,z,-3);
F2 = Y2*(z^(n-1))*(z+3);
y2 = limit(F2,z,-3);
F3 = Y2*(z^(n-1))*(z-1);
y3 = limit(F3,z,1);
disp((y1+y2+y3)*'u(n)','y[n]=')
//Result
//y[n] =  (2-(-3)^(n+1))*u(n)
```

Scilab code Exa 10.37

Output response of an LTI System

```
//clear//
//Example 10.37:To find output response of an LTI System
syms n z;
alpha = 8; //input constant
```

```
beta_b = 1; //initial condition y[-1] = 1
Y1 = -((3*beta_b*z)/(z+3))
Y2 = (alpha*z^2/((z+3)*(z-1)))
F1 = Y1*(z^(n-1))*(z+3);
y1 = limit(F1,z,-3);
F2 = Y2*(z^(n-1))*(z+3);
y2 = limit(F2,z,-3);
F3 = Y2*(z^(n-1))*(z-1);
y3 = limit(F3,z,1);
disp((y1+y2+y3)*'u(n)','y[n]=')
//Result
//y[n] =  (2-(-3)^(n+1))*u(n)
```

Chapter 12

Linear Feedback Systems

Chapter 13

Linear Feedback Systems

Scilab code Exa 11.1 Root locus Analysis of Linear Feedback Systems

```
//clear//
//Example11.1:Root locus Analysis of Linear Feedback
    Systems
//Continuous Time Systems
//Refer figure 11.12(a) in Openhiem &Willksy page
   840
s = %s;
H = syslin('c',[1/(s+1)]);
G = syslin('c',[1/(s+2)]);
F = G*H;
clf;
evans(F,3)
```

Scilab code Exa 11.1

Root locus Analysis of Linear Feedback Systems

```
//clear//
//Example11.1:Root locus Analysis of Linear Feedback
    Systems
```

```
//Continuous Time Systems
//Refer figure 11.12(a) in Openhiem &Willksy page 840
s = %s;
H = syslin('c',[1/(s+1)]);
G = syslin('c',[1/(s+2)]);
F = G*H;
clf;
evans(F,3)
```

Scilab code Exa 11.2 Continuous Time Systems

```
//clear//
//Example11.2:Root locus Analysis of Linear Feedback Systems
//Continuous Time Systems
//Refer figure 11.14(a) in Openhiem &Willksy page 844
s = %s;
G = syslin('c',[(s-1)/((s+1)*(s+2))]);
clf;
evans(G,2)
```

Scilab code Exa 11.2

Continuous Time Systems

```
//clear//
//Example11.2:Root locus Analysis of Linear Feedback Systems
//Continuous Time Systems
//Refer figure 11.14(a) in Openhiem &Willksy page 844
s = %s;
```

```
G = syslin('c',[(s-1)/((s+1)*(s+2))]);
clf;
evans(G,2)
```

Scilab code Exa 11.3 Discrete time system

```
//clear//
//Example11.3:Root locus Analysis of Linear Feedback
    Systems
////Discrete time system
//Refer figure 11.16(a) in Openhiem &Willksy page
   846
z = %z;
G = syslin('d',[z/((z-0.5)*(z-0.25))]);
clf;
evans(G,2)
```

Scilab code Exa 11.3

Discrete time system

```
//clear//
//Example11.3:Root locus Analysis of Linear Feedback
    Systems
////Discrete time system
//Refer figure 11.16(a) in Openhiem &Willksy page
   846
z = %z;
G = syslin('d',[z/((z-0.5)*(z-0.25))]);
clf;
evans(G,2)
```

Scilab code Exa 11.05 Nyquist criterion for Continuous Time Systems

```
//clear//
//Example 11.5:Nyquist criterion for Continuous Time
    Systems
//Nyquist Plot
s = %s;
//Open Loop Transfer Function
G = syslin('c',[1/(s+1)]);
H = syslin('c',[1/(0.5*s+1)]);
F = G*H;
clf;
nyquist(F)
show_margins(F,'nyquist')
```

Scilab code Exa 11.05

Nyquist criterion for Continuous Time Systems

```
//clear//
//Example 11.5:Nyquist criterion for Continuous Time
    Systems
//Nyquist Plot
s = %s;
//Open Loop Transfer Function
G = syslin('c',[1/(s+1)]);
H = syslin('c',[1/(0.5*s+1)]);
F = G*H;
clf;
nyquist(F)
show_margins(F,'nyquist')
```

Scilab code Exa 11.5 Bode Plot

```
//clear//
//Example 11.5:Nyquist criterion for Continuous Time
    Systems
//Bode Plot
s = %s;
//Open Loop Transfer Function
G = syslin('c',[1/(s+1)]);
H = syslin('c',[1/(0.5*s+1)]);
F = G*H;
clf;
bode(F,0.01,100)
show_margins(F)
```

Scilab code Exa 11.5

Bode Plot

```
//clear//
//Example 11.5:Nyquist criterion for Continuous Time
    Systems
//Bode Plot
s = %s;
//Open Loop Transfer Function
G = syslin('c',[1/(s+1)]);
H = syslin('c',[1/(0.5*s+1)]);
F = G*H;
clf;
bode(F,0.01,100)
show_margins(F)
```

Scilab code Exa 11.6 Nyquist Plot

```
//clear//
//Example 11.6:Nyquist criterion for Continuous Time
    Systems
//Nyquist Plot
s = %s;
//Open Loop Transfer Function
F = syslin('c',[(s+1)/((s-1)*(0.5*s+1))])
clf;
nyquist(F)
show_margins(F,'nyquist')
```

Scilab code Exa 11.6 Nyquist Plot

```
//clear//
//Example 11.6:Nyquist criterion for Continuous Time
    Systems
//Nyquist Plot
s = %s;
//Open Loop Transfer Function
F = syslin('c',[(s+1)/((s-1)*(0.5*s+1))])
clf;
nyquist(F)
show_margins(F,'nyquist')
```

Scilab code Exa 11.7 Nyquist Plot

```
//clear//
//Example 11.7
//Nyquist Plot
```

```
s = %s;
T =1;
//Open Loop Transfer Function
G = syslin('c',[-%e^(-s*T)]);
clf;
nyquist(G)
show_margins(G,'nyquist')
```

Scilab code Exa 11.7

Nyquist Plot

```
//clear//
//Example 11.7
//Nyquist Plot
s = %s;
T =1;
//Open Loop Transfer Function
G = syslin('c',[-%e^(-s*T)]);
clf;
nyquist(G)
show_margins(G,'nyquist')
```

Scilab code Exa 11.8 Nyquist Plot

```
//clear//
//Example 11.8:Nyquist criterion for Discrete Time
    Systems
//Nyquist Plot
//Discrete Time System
z = %z;
//Open Loop Transfer Function
F = syslin('d',[1/(z*(z+0.5))])
clf;
```

```
nyquist(F)
show_margins(F,'nyquist')
```

Scilab code Exa 11.8

Nyquist Plot

```
//clear//
//Example 11.8:Nyquist criterion for Discrete Time
    Systems
//Nyquist Plot
//Discrete Time System
z = %z;
//Open Loop Transfer Function
F = syslin('d',[1/(z*(z+0.5))])
clf;
nyquist(F)
show_margins(F,'nyquist')
```

Scilab code Exa 11.09 Root locus analysis of Linear feedback systems

```
//clear//
//Figure11.9:Root locus analysis of Linear feedback
    systems
s = %s;
beta_b1 = 1;
beta_b2 = -1;
G1 = syslin('c',[2*beta_b1/s]);
G2 = syslin('c',[2*beta_b2/s]);
H = syslin('c',[s/(s-2)]);
F1 = G1*H;
F2 = G2*H;
clf;
evans(F1,2)
```

```
figure
evans(F2,2)
```

Scilab code Exa 11.09

Root locus analysis of Linear feedback systems

```
//clear//
//Figure11.9:Root locus analysis of Linear feedback
   systems
s = %s;
beta_b1 = 1;
beta_b2 = -1;
G1 = syslin('c',[2*beta_b1/s]);
G2 = syslin('c',[2*beta_b2/s]);
H = syslin('c',[s/(s-2)]);
F1 = G1*H;
F2 = G2*H;
clf;
evans(F1,2)
figure
evans(F2,2)
```

Scilab code Exa 11.9 Gain and Phase Margins

```
//clear//
//Example 11.9:Gain and Phase Margins and their
//associated cross over frequencies
s =poly(0,'s'); // Define ss as polynomial variable
//Create s transfer function in forward path
F = syslin('c',[(4*(1+0.5*s))/(s*(1+2*s)*(1+0.05*s
   +(0.125*s)^2))])
B = syslin('c',(1+0*s)/(1+0*s))
OL = F*B;
```

```
fmin = 0.01; // Min freq in Hz
fmax = 10; // Max freq in Hz
scf(1);
//clf;
// Plot frequency response of open loop transfer
   function
bode(OL,0.01,10);
// display gain and phase margin and cross over
   frequencies
show_margins(OL);
[gm,fr1] = g_margin(OL)
[phm,fr2] = p_margin(OL)
disp(gm,'gain margin in dB')
disp(fr1,'gain cross over frequency in Hz')
disp(phm,'phase margin in dB')
disp(fr2,'phase cross over frequency in Hz')
```

Scilab code Exa 11.9 Gain and Phase Margins

```
//clear//
//Example 11.9:Gain and Phase Margins and their
//associated cross over frequencies
s =poly(0,'s'); // Define ss as polynomial variable
//Create s transfer function in forward path
F = syslin('c',[(4*(1+0.5*s))/(s*(1+2*s)*(1+0.05*s
   +(0.125*s)^2))])
B = syslin('c',(1+0*s)/(1+0*s))
OL = F*B;
fmin = 0.01; // Min freq in Hz
fmax = 10; // Max freq in Hz
scf(1);
//clf;
// Plot frequency response of open loop transfer
   function
bode(OL,0.01,10);
// display gain and phase margin and cross over
   frequencies
show_margins(OL);
```

```
[gm,fr1] = g_margin(OL)
[phm,fr2] = p_margin(OL)
disp(gm,'gain margin in dB')
disp(fr1,'gain cross over frequency in Hz')
disp(phm,'phase margin in dB')
disp(fr2,'phase cross over frequency in Hz')
```