

Scilab Manual for
DIGITAL SIGNAL PROCESSING
by Prof Desai Siddhibahen Deveshbhai
Others
Laxmi Institute Of Technology,sarigam¹

Solutions provided by
Prof Desai Siddhibahen Deveshbhai
Others
Laxmi Institute Of Technology,sarigam

June 2, 2025

¹Funded by a grant from the National Mission on Education through ICT, <http://spoken-tutorial.org/NMEICT-Intro>. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website <http://scilab.in>

Contents

List of Scilab Solutions	4
1 WRITE A SCILAB PROGRAM TO GENERATE COMMON DISCRETE TIME SIGNALS.	7
2 WRITE A SCILAB PROGRAM TO OBSERVE THE EFFECTS OF LOWER SAMPLING RATE AND HIGHER SAMPLING RATE ON C.T. SIGNAL(SAMPLING THM.)	11
3 WRITE A SCILAB PROGRAM TO COMPUTE LINEARITY PROPERTY OF A GIVEN SIGNAL.	15
4 WRITE A SCILAB PROGRAM TO COMPUTE LINEAR CONVOLUTION OF TWO SEQUENCES USING BASIC EQUATION.	18
5 WRITE A SCILAB PROGRAM TO FIND AUTOCORRELATION AND CROSS CORRELATION OF THE GIVEN SEQUENCES.	22
6 WRITE A SCILAB PROGRAM TO FIND N-POINT DFT OF THE GIVEN SEQUENCE.	26
7 WRITE A SCILAB PROGRAM TO COMPUTE LINEAR CONVOLUTION OF TWO SEQUENCES USING DFT BASED APPROACH.	31
8 WRITE A SCILAB PROGRAM TO COMPUTE CIRCULAR CONVOLUTION OF TWO SEQUECNES USING BA-	

SIC EQUATION.	34
9 WRITE A SCILAB PROGRAM TO COMPUTE CIRCULAR CONVOLUTION OF THE TWO SEQUENCES USING DFT BASED APPROACH.	38
10 WRITE A SCILAB PROGRAM TO COMPUTE BLOCK CONVOLUTION USING OVERLAP ADD METHOD	42
11 WRITE A SCILAB PROGRAM RO COMPUTE BLOCK CONVOLUTION USING OVERLAP SAVE METHOD.	46
12 WRITE A SCILAB PROGRAM TO FIND FFT USING DECIMATION IN TIME(DIT) METHOD.	50
13 WRITE A SCILAB PROGRAM FOR DESIGNING OF FIR FILTER FOR LOW PASS, HIGH PASS, BANDPASS AND BAND REJECT RESPONSES.	54
14 WRITE A SCILAB PROGRAM TO DESIGN DIGITAL IIR BUTTERWORTH LOW PASS FILTER.	62
15 WRITE A SCILAB PROGRAM TO DESIGN DIGITAL IIR CHEBYSHEW FILTER.	64

List of Experiments

Solution 1.01	PROGRAM TO GENERATE COMMON DISCRETE TIME SIGNALS	7
Solution 2.02	PROGRAM TO OBSERVE THE EFFECTS OF LOWER SAMPLING RATE AND HIGHER SAMPLING RATE ON CONTINUOUS TIME SIGNAL	11
Solution 3.03	PROGRAM TO COMPUTE LINEARITY PROPERTY OF GIVEN SIGNAL	15
Solution 4.04	PROGRAM TO COMPUTE LINEAR CONVOLUTION OF TWO SEQUENCES USING BASIC EQUATION	18
Solution 5.05	PROGRAM TO FIND AUTO CORRELATION AND CROSS CORRELATION OF THE GIVEN SEQUENCES	22
Solution 6.06	PROGRAM TO FIND N POINT DFT OF THE GIVEN SEQUENCE	26
Solution 7.07	PROGRAM TO COMPUTE LINEAR CONVOLUTION OF TWO SEQUENCES USING DFT BASED APPROACH	31
Solution 8.08	PROGRAM TO COMPUTE CIRCULAR CONVOLUTION OF TWO SEQUENCES USING BASIC EQUATION	34
Solution 9.09	PROGRAM TO COMPUTE CIRCULAR CONVOLUTION OF THE TWO SEQUENCES USING DFT BASED APPROACH	38
Solution 10.10	PROGRAM TO COMPUTE BLOCK CONVOLUTION USING OVERLAP ADD METHOD	42
Solution 11.11	PROGRAM TO COMPUTE BLOCK CONVOLUTION USING OVERLAP SAVE METHOD	46

Solution 12.12	PROGRAM TO FIND FFT USING DECIMATION IN TIME METHOD	50
Solution 13.13	PROGRAM FOR DESIGNING OF FIR FILTER FOR LOW PASS HIGH PASS BAND PASS AND BAND REJECT RESPONSES	54
Solution 14.14	PROGRAM TO DESIGN DIGITAL IIR BUTTER- WORTH LOW PASS FILTER	62
Solution 15.15	PROGRAM TO DESIGN DIGITAL IIR CHEBYSHEW FILTER	64

List of Figures

Experiment: 1

WRITE A SCILAB PROGRAM TO GENERATE COMMON DISCRETE TIME SIGNALS.

Scilab code Solution 1.01 PROGRAM TO GENERATE COMMON DIS-
CRETE TIME SIGNALS

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 // CAPTION: PROGRAM TO GENERATE COMMON DISCRETE TIME
  SIGNALS
4
5 clc;
6 clear;
7 close;
8
9 //UNIT IMPULSE SIGNAL
10 L=input('Enter the Length='); //SET LIMIT
11 n=-L:L;
```



```

12 x=[zeros(1,L),ones(1,1),zeros(1,L)];
13 subplot(2,4,1);
14 a=gca();
15 a.y_location =" origin";
16 a.foreground = 5;
17 a.font_color = 5;
18 a.font_style = 5;
19 plot2d3(n,x)
20 title('Unit Impulse Signal');
21 xlabel('Samples n');
22 ylabel('Amplitude');
23
24 // UNIT STEP SIGNAL
25 y=[zeros(1,L),ones(1,L+1)];
26 subplot(3,4,2);
27 b=gca();
28 b.y_location =" middle";
29 b.foreground = 5;
30 b.font_color = 5;
31 b.font_style = 5;
32 plot2d3(n,y)
33 title('Unit Step Signal');
34 xlabel('Samples n');
35 ylabel('Amplitude');
36
37 //UNIT RAMP SIGNAL
38 z=[zeros(1,L),0:L],
39 subplot(2,4,3)
40 c=gca();
41 c.y_location =" middle";
42 c.foreground = 5;
43 c.font_color = 5;
44 c.font_style = 5;
45 plot2d3(n,z)
46 title('Unit Ramp Signal');
47 xlabel('Samples n');
48 ylabel('Amplitude');
49

```

```

50 // EXPONENTIALLY INCREASING SEQUENCE
51
52 n=0:1:10;
53 x=exp(n);
54 subplot(2,4,4);
55 d=gca();
56 d.x_location="origin";
57 d.y_location="origin";
58 d.foreground = 5;
59 d.font_color = 5;
60 d.font_style = 5;
61 plot2d3(n,x)
62 title('Exponentially Increasing Sequence');
63 xlabel('Samples n');
64 ylabel('Amplitude');
65
66 // EXPONENTIALLY DECREASING SEQUENCE
67 n=0:1:10;
68 x=exp(-n);
69 subplot(2,4,5);
70 d=gca();
71 d.x_location="origin";
72 d.y_location="origin";
73 d.foreground = 5;
74 d.font_color = 5;
75 d.font_style = 5;
76 plot2d3(n,x)
77 title('Exponentially Decreasing Sequence');
78 xlabel('Samples n');
79 ylabel('Amplitude');
80
81 // SINE WAVE
82 t=0:0.04:1;
83 x=sin(2*pi*t);
84 subplot(2,4,6);
85 a=gca();
86 a.foreground = 5;
87 a.font_color = 5;

```

```

88 a.font_style = 5;
89 plot2d3(t,x);
90 title('Sine Wave')
91 xlabel('Samples n');
92 ylabel('Amplitude');
93
94 // COSINE WAVE
95 t=0:0.04:1;
96 x=cos(2*%pi*t);
97 subplot(2,4,7);
98 b=gca();
99 b.foreground = 5;
100 b.font_color = 5;
101 b.font_style = 5;
102 plot2d3(t,x);
103 title('Cosine Wave')
104 xlabel('Samples n');
105 ylabel('Amplitude');
106
107 //INPUT:
108 //Enter the Length=5

```

Experiment: 2

WRITE A SCILAB
PROGRAM TO OBSERVE
THE EFFECTS OF LOWER
SAMPLING RATE AND
HIGHER SAMPLING RATE
ON C.T. SIGNAL(SAMPLING
THM.)

Scilab code Solution 2.02 PROGRAM TO OBSERVE THE EFFECTS
OF LOWER SAMPLING RATE AND HIGHER SAMPLING RATE ON
CONTINUOUS TIME SIGNAL

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 //CAPTION: PROGRAM TO OBSERVE THE EFFECTS OF LOWER
  SAMPLING RATE AND
```

```

4 // HIGHER SAMPLING RATE ON C.T. SIGNAL(SAMPLING THM
   .)
5
6 clc;
7 clear;
8 close;
9 f=input('Enter continuous time signal frequency=');
   // f=0.1
10 a=input('Enter continuous time signal amplitude=');
   //a=1
11 t=0:0.01:100;
12 x=a*sin(2*%pi*f*t);
13 subplot(2,2,1);
14 b=gca();
15 b.x_location="origin";
16 b.y_location="origin";
17 b.foreground = 5;
18 b.font_color = 5;
19 b.font_style = 5;
20 plot(t,x);
21 title('Original Continuous time Signal');
22 xlabel('Time t');
23 ylabel('Amplitude');
24
25 // Sampling Rate=2*max frequency
26 fs1=input('Enter sampling frequency equal to 2*fs=')
   ;//fs1=0.2
27 fd=f/fs1;
28 n=0:0.01:100;
29 x1n=a*sin(2*%pi*f*n/fs1);
30 subplot(2,2,2);
31 b=gca();
32 b.x_location="origin";
33 b.y_location="origin";
34 b.foreground = 5;
35 b.font_color = 5;
36 b.font_style = 5;
37 plot2d3('gnn',n,x1n);

```

```

38 title('Reconstructed signal with sampling frequency
    equal 2*fs ');
39 xlabel('Time t ');
40 ylabel('Amplitude');
41
42
43 // Sampling Rate<2*max frequency= Aliasing Effect
44 fs2=input('Enter sampling frequency less than 2*fs='
    ');//fs2=0.1
45 x2n=a*sin(2*%pi*f*n/fs2);
46 subplot(2,2,3);
47 b=gca();
48 b.x_location="origin";
49 b.y_location="origin";
50 b.foreground = 5;
51 b.font_color = 5;
52 b.font_style = 5;
53 plot2d3('gnn',n,x2n);
54 title('Reconstructed signal with sampling frequency
    Less than 2*fs=Aliasing effect ');
55 xlabel('Time t ');
56 ylabel('Amplitude');
57
58 // Sampling Rate>2*max frequency=Perfect
    Reconstuection
59 fs3=input('Enter sampling frequency Greater than 2*
    fs=');//fs3=1
60 x3n=a*sin(2*%pi*f*n*fs3);
61 subplot(2,2,4);
62 b=gca();
63 b.x_location="origin";
64 b.y_location="origin";
65 b.foreground = 5;
66 b.font_color = 5;
67 b.font_style = 5;
68 plot2d3('gnn',n,x3n);
69 title('Perfect Reconstructed signal with sampling
    frequency greater than 2*fs ');

```

```
70 xlabel('Time t');
71 ylabel('Amplitude');
72
73 //INPUT:
74 //Enter continuous time signal frequency=0.1
75 //Enter continuous time signal amplitude=1
76 //Enter sampling frequency equal to 2*fs=0.2
77 //Enter sampling frequency less than 2*fs=0.1
78 //Enter sampling frequency Greater than 2*fs=1
```

Experiment: 3

WRITE A SCILAB PROGRAM TO COMPUTE LINEARITY PROPERTY OF A GIVEN SIGNAL.

Scilab code Solution 3.03 PROGRAM TO COMPUTE LINEARITY PROPERTY OF GIVEN SIGNAL

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 // CAPTION: PROGRAM TO COMPUTE LINEARITY PROPERTY OF
  A GIVEN SIGNAL
4 //Input Sequence:    x1(n)=sin(2%pi0.1n) and x2(n)=
  cos(2%pi0.5n) and system y(n)=0.5x(n)
5 clc;
6 clear;
7 close;
8 n=0:10;
9 a=2;
10 b=3;
```



```

11 x1=sin(2*%pi*0.1*n); // input sequence 1
12 x2=cos(2*%pi*0.5*n); //input sequence 2
13 x=a.*x1+b.*x2; // homogeneity and superposition —op1
14 y=0.5*x; // system y=0.5*x
15 y1=0.5*x1;
16 y2=0.5*x2;
17 yt=a.*y1+b.*y2; // homogeneity and superposition —op2
18 d=y-yt; // op2-op1=difference
19 disp('Output of System y(n)=0.5x(n) is:');
20 if(d==0)
21     disp('System is linear');
22 else
23     disp('System is Non-linear');
24 end
25
26 //Input Sequence: x1(n)=sin(2%pi0.1n) and x2(n)=
    cos(2%pi0.5n) and system y(n)=sqrt(x(n))
27 n=0:10;
28 a=2;
29 b=3;
30 x1=sin(2*%pi*0.1*n); //input sequence 1
31 x2=cos(2*%pi*0.5*n); //input sequence 2
32 x=a.*x1+b.*x2; // homogeneity and superposition —op1
33 y=sqrt(x); // system y=sqrt(x)
34 y1=sqrt(x1);
35 y2=sqrt(x2);
36 yt=a.*y1+b.*y2; // homogeneity and superposition —op2
37 d=y-yt; // op2-op1=difference
38 disp('Output of System y(n)=sqrt(x(n)) is:');
39 if(d==0)
40     disp('System is linear');
41 else
42     disp('System is Non-linear');
43 end
44
45
46 //OUTPUT:
47 //Output of System y(n)=0.5x(n) is:

```

```
48
49 //System is linear
50
51 //Output of System  $y(n)=\sqrt{x(n)}$  is :
52
53 // System is Non-linear
```

Experiment: 4

WRITE A SCILAB PROGRAM TO COMPUTE LINEAR CONVOLUTION OF TWO SEQUENCES USING BASIC EQUATION.

Scilab code Solution 4.04 PROGRAM TO COMPUTE LINEAR CONVOLUTION OF TWO SEQUENCES USING BASIC EQUATION

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 //CAPTION: PROGRAM TO COMPUTE LINEAR CONVOLUTION OF
  TWO SEQUENCES USING BASIC EQUATION
4 clc;
5 clear;
6 close;
7 //input sequences
8 x=input('Enter the Input Sequence x(n)=') //x=[1 2 3
  1]
```

```

9  m=length(x);
10 x1=input('Enter the lower index of Input Sequence=')
    //0
11 xh=x1+m-1;
12 n=x1:1:xh;
13 subplot(3,1,1);
14 a=gca();
15 a.x_location="origin";
16 a.y_location="origin";
17 a.foreground = 5;
18 a.font_color = 5;
19 a.font_style = 5;
20 plot2d3('gnn',n,x);
21 title('Input Sequence x[n]');
22 xlabel('Samples n');
23 ylabel('Amplitude');
24
25 h=input('Enter the Impulse response Sequence h(n)=')
    ;//h=[1 2 1 -1]
26 l=length(h);
27 h1=input('Enter the lower index of Impulse response
    Sequence=');//-1
28 hh=h1+l-1;
29 g=h1:1:hh;
30 subplot(3,1,2);
31 a=gca();
32 a.x_location="origin";
33 a.y_location="origin";
34 a.foreground = 5;
35 a.font_color = 5;
36 a.font_style = 5;
37 plot2d3('gnn',n,h);
38 title('Impulse Response Sequence h[n]');
39 xlabel('Samples n');
40 ylabel('Amplitude');
41
42 nx=x1+h1; // range of k
43 nh=xh+hh; // range of n

```

```

44 x=[x,zeros(1,1)];
45 h=[h,zeros(1,m)];
46 y=zeros(1,m+1-1) // n1+n2-1=length of linear
    convolution
47 for i=1:m+1-1
48     y(i)=0;
49     for j=1:m+1-1
50         if(j<i+1)
51             y(i)=y(i)+x(j)*h(i-j+1);
52         end
53     end
54 end
55 disp('Linear Convolution using Equation is y(n):')
56 disp(y);
57 r=nx:nh;
58 subplot(3,1,3);
59 a=gca();
60 a.x_location="origin";
61 a.y_location="origin";
62 a.foreground = 5;
63 a.font_color = 5;
64 a.font_style = 5;
65 plot2d3('gnn',r,y);
66 title('Output Response Sequence of Linear
    Convolution using Equation y[n]');
67 xlabel('Samples n');
68 ylabel('Amplitude');
69 //INPUT:
70 //Enter the Input Sequence x(n)=[1 2 3 1]
71 //Enter the lower index of Input Sequence=0
72 //Enter the Impulse response Sequence h(n)=[1 2 1
    -1]
73 //Enter the lower index of Impulse response Sequence
    =-1
74
75 //OUTPUT:
76 //Linear Convolution using Equation is y(n):
77

```

78 // 1. 4. 8. 8. 3. – 2. – 1.

Experiment: 5

WRITE A SCILAB PROGRAM TO FIND AUTOCORRELATION AND CROSS CORRELATION OF THE GIVEN SEQUENCES.

Scilab code Solution 5.05 PROGRAM TO FIND AUTO CORRELATION
AND CROSS CORRELATION OF THE GIVEN SEQUENCES

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 //CAPTION: PROGRAM TO FIND AUTOCORRELATION AND
  CROSSCORRELATION OF THE GIVEN SEQUENCES
4 clc;
5 clear;
6 close;
7 x=input('Enter the Input Sequence=') //x=[1 2 3 1]
8 m=length(x);
9 x1=input('Enter the lower index of Input Sequence=')
```

```

        //0
10  xh=xl+m-1;
11  n=xl:1:xh;
12  subplot(2,2,1);
13  a=gca();
14  a.x_location="origin";
15  a.y_location="origin";
16  a.foreground = 5;
17  a.font_color = 5;
18  a.font_style = 5;
19  plot2d3('gnn',n,x);
20  title('Input Sequence x[n]');
21  xlabel('Samples n');
22  ylabel('Amplitude');
23
24  h=input('Enter the Impulse response Sequence='); //h
    =[1 2 1 1]
25  l=length(h);
26  hl=input('Enter the lower index of impulse response
    Sequence='); //0
27  hh=hl+l-1;
28  g=hl:1:hh;
29  subplot(2,2,2);
30  a=gca();
31  a.x_location="origin";
32  a.y_location="origin";
33  a.foreground = 5;
34  a.font_color = 5;
35  a.font_style = 5;
36  plot2d3('gnn',g,h);
37  title('Impulse Response Sequence h[n]');
38  xlabel('Samples n');
39  ylabel('Amplitude');
40
41  // Autocorrelation
42  y=xcorr(x,x);
43  disp('Auto Correlation Of given Sequence y(n)=');
44  disp(y);

```



```

45 nx=xl+xl;
46 nh=xh+xh;
47 r=nx:nh;
48 subplot(2,2,3);
49 a=gca();
50 a.x_location="origin";
51 a.y_location="origin";
52 a.foreground = 5;
53 a.font_color = 5;
54 a.font_style = 5;
55 plot2d3('gnn',r,y);
56 title('Output of Auto Correlation of Sequence');
57 xlabel('Samples n');
58 ylabel('Amplitude');
59
60 //Cross correlation
61 z=xcorr(x,h);
62 disp('Cross Correlation of Sequence y(n)=');
63 disp(z);
64 subplot(2,2,4);
65 a=gca();
66 a.x_location="origin";
67 a.y_location="origin";
68 a.foreground = 5;
69 a.font_color = 5;
70 a.font_style = 5;
71 plot2d3('gnn',r,z);
72 title('Output of Cross correlation of Sequence');
73 xlabel('Samples n');
74 ylabel('Amplitude');
75
76 //INPUT:
77 //Enter the Input Sequence=[1 2 3 1]
78 //Enter the lower index of Input Sequence=0
79 //Enter the Impulse response Sequence=[1 2 1 1]
80 //Enter the lower index of impulse response Sequence
    =0
81

```

```
82 //OUTPUT:
83 //Auto Correlation Of given Sequence y(n)=
84
85 // 1.    5.    11.    15.    11.    5.    1.
86
87 //Cross Correlation of Sequence y(n)=
88
89 // 1.    3.    7.    9.    9.    5.    1.
```

Experiment: 6

WRITE A SCILAB PROGRAM TO FIND N-POINT DFT OF THE GIVEN SEQUENCE.

Scilab code Solution 6.06 PROGRAM TO FIND N POINT DFT OF
THE GIVEN SEQUENCE

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 // CAPTION: PROGRAM TO FIND N-POINT DFT OF THE GIVEN
  SEQUENCE
4 clc;
5 clear;
6 close;
7 N=input('Enter the value of N='); //n-point
8 x=input('Enter the input sequence x(n)=');
9 subplot(3,2,1);
10 a=gca();
11 a.foreground = 5;
```

```

12 a.font_color = 5;
13 a.font_style = 5;
14 plot2d3(x);
15 title('Input sequence');
16 xlabel('Samples n');
17 ylabel('Amplitude');
18
19 //calculation of DFT
20 for k=1:N
21     y(k)=0;
22     for n=1:N
23         y(k)=y(k)+x(n).*exp(-%i*2*%pi*(k-1)*(n-1)/N)
24         ;
25         A=real(y);
26         B=imag(y);
27     end
28 mag=abs(y);
29 x1=atan(imag(y),real(y));
30 phase=x1*(180/%pi);
31 disp('The output DFT sequence is:');
32 disp(y);
33 subplot(3,2,2);
34 a=gca();
35 a.foreground = 5;
36 a.font_color = 5;
37 a.font_style = 5;
38 plot2d3(y);
39 title('Output DFT sequence');
40 xlabel('Samples n');
41 ylabel('Amplitude');
42
43 //REAL VALUE
44 disp('The resultant real value is:');
45 disp(A);
46 subplot(3,2,3);
47 a=gca();
48 a.foreground = 5;

```

```

49 a.font_color = 5;
50 a.font_style = 5;
51 plot2d3(A);
52 title('Real Value');
53 xlabel('Samples n');
54 ylabel('Amplitude');
55
56 //IMAGINARY VALUE
57 disp('The resultant imaginary value is:');
58 disp(B);
59 subplot(3,2,4);
60 a=gca();
61 a.foreground = 5;
62 a.font_color = 5;
63 a.font_style = 5;
64 plot2d3(B);
65 title('Imaginary Value');
66 xlabel('Samples n');
67 ylabel('Amplitude');
68
69 //MAGNITUDE RESPONSE
70 disp('The Magnitude response is:');
71 disp(mag);
72 subplot(3,2,5);
73 a=gca();
74 a.foreground = 5;
75 a.font_color = 5;
76 a.font_style = 5;
77 plot2d3(mag);
78 title('Magnitude Response');
79 xlabel('Samples n');
80 ylabel('Amplitude');
81
82 //PHASE RESPONSE
83 disp('The Phase response is:');
84 disp(phase);
85 subplot(3,2,6);
86 a=gca();

```

```

87 a.foreground = 5;
88 a.font_color = 5;
89 a.font_style = 5;
90 plot2d3(phase);
91 title('Phase Response');
92 xlabel('Samples n');
93 ylabel('Phase');
94
95
96
97 //INPUT:
98 //Enter the value of N=4
99 //Enter the input sequence x(n)=[1 2 3 4]
100
101 //OUTPUT:
102 //The output DFT sequence is:
103
104 // 10.
105 //- 2. + 2.i
106 //- 2. - 9.797D-16i
107 //- 2. - 2.i
108
109 //The resultant real value is:
110
111 // 10.
112 //- 2.
113 //- 2.
114 //- 2.
115
116 //The resultant imaginary value is:
117
118 // 0.
119 // 2.
120 //- 9.797D-16
121 //- 2.
122
123 //The Magnitude response is:
124

```

```
125      // 10.  
126      // 2.8284271  
127      // 2.  
128      // 2.8284271  
129  
130  //The Phase response is :  
131  
132      // 0.  
133      // 135.  
134      // - 180.  
135      // - 135.
```

Experiment: 7

WRITE A SCILAB PROGRAM TO COMPUTE LINEAR CONVOLUTION OF TWO SEQUENCES USING DFT BASED APPROACH.

Scilab code Solution 7.07 PROGRAM TO COMPUTE LINEAR CONVOLUTION OF TWO SEQUENCES USING DFT BASED APPROACH

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 //CAPTION: PROGRAM TO COMPUTE LINEAR CONVOLUTION OF
  TWO SEQUENCES USING DFT BASED APPROACH
4 clc;
5 clear;
6 close;
7 x=input('Enter the input sequence x(n)=') //x=[1 2 3
    1]
8 m=length(x);
```



```

9  x1=input('Enter the lower index of input sequence=')
    //0
10  xh=x1+m-1;
11  n=x1:1:xh;
12  subplot(3,1,1);
13  a=gca();
14  a.x_location="origin";
15  a.y_location="origin";
16  a.foreground = 5;
17  a.font_color = 5;
18  a.font_style = 5;
19  plot2d3('gnn',n,x);
20  title('Input Sequence x[n]');
21  xlabel('Samples n');
22  ylabel('Amplitude');
23
24  h=input('Enter the impulse response sequence h(n)=')
    ;//h=[1 2 1 -1]
25  l=length(h);
26  h1=input('Enter the lower index of impulse response
    sequence='); //-1
27  hh=h1+l-1;
28  g=h1:1:hh;
29  subplot(3,1,2);
30  a=gca();
31  a.x_location="origin";
32  a.y_location="origin";
33  a.foreground = 5;
34  a.font_color = 5;
35  a.font_style = 5;
36  plot2d3('gnn',n,h);
37  title('Impulse Response Sequence h[n]');
38  xlabel('Samples n');
39  ylabel('Amplitude');
40
41  nx=x1+h1; // range of k
42  nh=xh+hh; // range of n
43  p=m+l-1;

```

```

44 x=[x,zeros(1,p-1)];
45 h=[h,zeros(1,p-m)];
46
47 // dft-idft
48 XK=fft(x,-1); // DFT of x
49 HK=fft(h,-1); // DFT of h
50 YK=XK.*HK; // Multiplication of DFT of x and h
51 yn=fft(YK,1); // Inverse DFT for obtaining yn
52 disp('Linear Convolution by DFT-IDFT Method is y(n):
      ');
53 disp(real(yn));
54 r=nx:nh;
55 subplot(3,1,3);
56 a=gca();
57 a.x_location="origin";
58 a.y_location="origin";
59 a.foreground = 5;
60 a.font_color = 5;
61 a.font_style = 5;
62 plot2d3('gnn',r,yn);
63 title('Output Response Sequence of Linear
      Convolution by DFT-IDFT Method y[n]');
64 xlabel('Samples n');
65 ylabel('Amplitude');
66
67 //INPUT:
68 //Enter the input sequence x(n)=[1 2 3 1]
69 //Enter the lower index of input sequence=0
70 //Enter the impulse response sequence h(n)=[1 2 1
      -1]
71 //Enter the lower index of impulse response sequence
      =-1
72
73 //OUTPUT:
74 //Linear Convolution by DFT-IDFT Method is y(n):
75
76 // 1.    4.    8.    8.    3.  - 2.  - 1.

```

Experiment: 8

WRITE A SCILAB PROGRAM TO COMPUTE CIRCULAR CONVOLUTION OF TWO SEQUECNES USING BASIC EQUATION.

Scilab code Solution 8.08 PROGRAM TO COMPUTE CIRCULAR CON-
VOLUTION OF TWO SEQUENCES USING BASIC EQUATION

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 //CAPTION: PROGRAM TO COMPUTE CIRCULAR CONVOLUTION
  OF TWO SEQUENCES USING BASIC EQUATION
4 clc;
5 clear;
6 close;
7 x=input('Enter the input sequence=') //x=[1 1 2 2]
8 m=length(x);
9 x1=input('Enter the lower index of input sequence=')
```

```

        //0
10  xh=xl+m-1;
11  n=xl:1:xh;
12  subplot(3,1,1);
13  a=gca();
14  a.x_location="origin";
15  a.y_location="origin";
16  a.foreground = 5;
17  a.font_color = 5;
18  a.font_style = 5;
19  plot2d3('gnn',n,x);
20  title('Input Sequence x[n]');
21  xlabel('Samples n');
22  ylabel('Amplitude');
23
24  h=input('Enter the impulse response sequence='); //h
    =[1 2 3 4]
25  l=length(h);
26  hl=input('Enter the lower index of impulse response
    sequence='); //0
27  hh=hl+l-1;
28  g=hl:1:hh;
29  subplot(3,1,2);
30  a=gca();
31  a.x_location="origin";
32  a.y_location="origin";
33  a.foreground = 5;
34  a.font_color = 5;
35  a.font_style = 5;
36  plot2d3('gnn',g,h);
37  title('Impulse Response Sequence h[n]');
38  xlabel('Samples n');
39  ylabel('Amplitude');
40
41
42  // for making length of both signals equal
43  N=max(m,l);
44  p=m-1;

```

```

45 if(p>=0) then
46     h=[h,zeros(1,p)];
47 else
48     x=[x,zeros(1,-p)];
49 end
50 for i=1:N
51     y(i)=0;
52     for j=1:N
53         k=i-j+1;
54         if(k<=0)
55             k=N+k;
56         end
57         y(i)=y(i)+x(j)*h(k);
58     end
59 end
60 disp(' Circular convolution by equation is y[n]:');
61 disp(y);
62 nx=xl+hl;
63 r=nx:length(y)-1;
64 subplot(3,1,3);
65 a=gca();
66 a.x_location="origin";
67 a.y_location="origin";
68 a.foreground = 5;
69 a.font_color = 5;
70 a.font_style = 5;
71 plot2d3('gnn',r,y);
72 title('Output Response Sequence of Circular
        Convolution y[n] using Basic Equation');
73 xlabel('Samples n');
74 ylabel('Amplitude');
75
76 //INPUT:
77 //Enter the input sequence=[1 1 2 2]
78 //Enter the lower index of input sequence=0
79 //Enter the impulse response sequence=[1 2 3 4]
80 //Enter the lower index of impulse response sequence
    =0

```

```
81
82 //OUTPUT:
83 //Circular convolution by equation is y[n]:
84
85 //      15.
86 //      17.
87 //      15.
88 //      13.
```

Experiment: 9

WRITE A SCILAB
PROGRAM TO COMPUTE
CIRCULAR CONVOLUTION
OF THE TWO SEQUENCES
USING DFT BASED
APPROACH.

Scilab code Solution 9.09 PROGRAM TO COMPUTE CIRCULAR CONVOLUTION OF THE TWO SEQUENCES USING DFT BASED APPROACH

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 //CAPTION:PROGRAM TO COMPUTE CIRCULAR CONVOLUTION OF
  THE TWO SEQUENCES USING DFT BASED APPROACH
4 clc;
5 clear;
6 close;
7 x=input('Enter the Input sequence=') //x=[1 1 2 2]
```

```

8 m=length(x);
9 x1=input('Enter the lower index of input sequence=')
  //0
10 xh=x1+m-1;
11 n=x1:1:xh;
12 subplot(3,1,1);
13 a=gca();
14 a.x_location="origin";
15 a.y_location="origin";
16 a.foreground = 5;
17 a.font_color = 5;
18 a.font_style = 5;
19 plot2d3('gnn',n,x);
20 title('Input Sequence x[n]');
21 xlabel('Samples n');
22 ylabel('Amplitude');
23
24 h=input('Enter the Impulse response sequence=');//h
  =[1 2 3 4]
25 l=length(h);
26 h1=input('Enter the lower index of impulse response
  sequence=');//0
27 hh=h1+l-1;
28 g=h1:1:hh;
29 subplot(3,1,2);
30 a=gca();
31 a.x_location="origin";
32 a.y_location="origin";
33 a.foreground = 5;
34 a.font_color = 5;
35 a.font_style = 5;
36 plot2d3('gnn',g,h);
37 title('Impulse Response Sequence h[n]');
38 xlabel('Samples n');
39 ylabel('Amplitude');
40
41 // for making length of both signals equal
42 N=max(m,l);

```



```

43 p=m-1;
44 if(p>=0) then
45     h=[h,zeros(1,p)];
46 else
47     x=[x,zeros(1,-p)];
48 end
49 XK=fft(x,-1);
50 HK=fft(h,-1);
51 YK=XK.*HK;
52 y=ifft(YK);
53 disp(' Circular convolution by DFT is y(n):');
54 disp(real(y));
55 nx=xl+hl;
56 r=nx:length(y)-1;
57 subplot(3,1,3);
58 a=gca();
59 a.x_location="origin";
60 a.y_location="origin";
61 a.foreground = 5;
62 a.font_color = 5;
63 a.font_style = 5;
64 plot2d3('gnn',r,y);
65 title('Output Response Sequence of Circular
        Convolution y[n] using DFT');
66 xlabel('Samples n');
67 ylabel('Amplitude');
68
69 //INPUT:
70 //Enter the Input sequence=[1 1 2 2]
71 //Enter the lower index of input sequence=0
72 //Enter the Impulse response sequence=[1 2 3 4]
73 //Enter the lower index of impulse response sequence
    =0
74
75 //OUTPUT:
76 //Circular convolution by DFT is y(n):
77
78 // 15.      17.      15.      13.

```


Experiment: 10

WRITE A SCILAB PROGRAM TO COMPUTE BLOCK CONVOLUTION USING OVERLAP ADD METHOD

Scilab code Solution 10.10 PROGRAM TO COMPUTE BLOCK CONVOLUTION USING OVERLAP ADD METHOD

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 //CAPTION: PROGRAM TO COMPUTE BLOCK CONVOLUTION
  USING OVERLAP ADD METHOD
4 clc;
5 clear;
6 close;
7 x=input('Enter the input sequence=') //x=[1 2 -1 2 3
    -2 -3 -1 1 1 2 -1]
8 m=length(x);
```

```

9  x1=input('Enter the lower index of input sequence=')
    //0
10  xh=x1+m-1;
11  n=x1:1:xh;
12  subplot(3,1,1);
13  a=gca();
14  a.x_location="origin";
15  a.y_location="origin";
16  a.foreground = 5;
17  a.font_color = 5;
18  a.font_style = 5;
19  plot2d3('gnn',n,x);
20  title('Input Sequence x[n]');
21  xlabel('Samples n');
22  ylabel('Amplitude');
23
24  h=input('Enter the impulse response sequence=');//h
    =[1 2 3 -1]
25  l=length(h);
26  h1=input('Enter the lower index of impulse response
    sequence=');//0
27  hh=h1+l-1;
28  g=h1:1:hh;
29  subplot(3,1,2);
30  a=gca();
31  a.x_location="origin";
32  a.y_location="origin";
33  a.foreground = 5;
34  a.font_color = 5;
35  a.font_style = 5;
36  plot2d3('gnn',g,h);
37  title('Impulse Response Sequence h[n]');
38  xlabel('Samples n');
39  ylabel('Amplitude');
40
41  N=m+l-1;
42  h1=[h zeros(1,l-1)];
43  n3=length(h1);

```

```

44 y=zeros(1,N);
45
46 H=fft(h1);
47 for i=1:l:m
48     if i<=(m+l-1) then
49         x1=[x(i:i+n3-1) zeros(1,n3-1)];
50     else
51         x1=[x(i:m) zeros(1,n3-1)];
52     end
53     x2=fft(x1);
54     x3=x2.*H;
55     x4=round(ifft(x3));
56     if(i==1)
57         y(1:n3)=x4(1:n3);
58     else
59         y(i:i+n3-1)=y(i:i+n3-1)+x4(1:n3);
60     end
61 end
62
63 disp('Output sequence using Overlap add method y(n):
        ');
64 disp(y(1:N));
65 nx=x1+h1;
66 r=nx:length(y)-1;
67 subplot(3,1,3);
68 a=gca();
69 a.x_location="origin";
70 a.y_location="origin";
71 a.foreground = 5;
72 a.font_color = 5;
73 a.font_style = 5;
74 plot2d3('gnn',r,y);
75 title('Output sequence using Overlap add method y[n]
        ');
76 xlabel('Samples n');
77 ylabel('Amplitude');
78
79 //INPUT:

```

```

80 //Enter the input sequence=[1 2 -1 2 3 -2 -3 -1 1 1
    2 -1]
81 //Enter the lower index of input sequence=0
82 //Enter the impulse response sequence=[1 2 3 -1]
83 //Enter the lower index of impulse response sequence
    =0
84
85 //OUTPUT:
86 //Output sequence using Overlap add method y(n):
87
88
89 //      column 1 to 12
90
91 //  1.      4.      6.      5.      2.      11.      0.  - 16.
    - 8.      3.      8.      5.
92
93 //      column 13 to 15
94
95 //  3.  - 5.      1.

```

Experiment: 11

WRITE A SCILAB PROGRAM TO COMPUTE BLOCK CONVOLUTION USING OVERLAP SAVE METHOD.

Scilab code Solution 11.11 PROGRAM TO COMPUTE BLOCK CONVOLUTION USING OVERLAP SAVE METHOD

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 //CAPTION: PROGRAM TO COMPUTE BLOCK CONVOLUTION
  USING OVERLAP SAVE METHOD
4 clc;
5 clear;
6 close;
7 x=input('Enter the input sequence=') //x=[1 2 -1 2 3
    -2 -3 -1 1 1 2 -1]
8 m=length(x);
```

```

9  x1=input('Enter the lower index of input sequence=')
    //0
10  xh=x1+m-1;
11  n=x1:1:xh;
12  subplot(3,1,1);
13  a=gca();
14  a.x_location="origin";
15  a.y_location="origin";
16  a.foreground = 5;
17  a.font_color = 5;
18  a.font_style = 5;
19  plot2d3('gnn',n,x);
20  title('Input Sequence x[n]');
21  xlabel('Samples n');
22  ylabel('Amplitude');
23
24  h=input('Enter the impulse response sequence=');//h
    =[1 2 3 -1]
25  l=length(h);
26  h1=input('Enter the lower index of impulse response
    sequence=');//0
27  hh=h1+l-1;
28  g=h1:1:hh;
29  subplot(3,1,2);
30  a=gca();
31  a.x_location="origin";
32  a.y_location="origin";
33  a.foreground = 5;
34  a.font_color = 5;
35  a.font_style = 5;
36  plot2d3('gnn',g,h);
37  title('Impulse Response Sequence h[n]');
38  xlabel('Samples n');
39  ylabel('Amplitude');
40
41  N=m+l-1;
42  h1=[h zeros(1,N-m)];
43  n3=length(h1);

```



```

44 y=zeros(1,N);
45 x1=[zeros(1,n3-1) x zeros(1,n3)];
46 H=fft(h1);
47 for i=1:1:N
48     y1=x1(i:i+(2*(n3-1)));
49     y2=fft(y1);
50     y3=y2.*H;
51     y4=round(ifft(y3));
52     y(i:(i+n3-1))=y4(1:n3);
53 end
54 disp('Output sequence using overlap save method Y(n)
      :');
55 disp(y(1:N));
56 nx=x1+h1;
57 r=nx:length(y)-1;
58 subplot(3,1,3);
59 a=gca();
60 a.x_location='origin';
61 a.y_location='origin';
62 a.foreground = 5;
63 a.font_color = 5;
64 a.font_style = 5;
65 plot2d3('gnn',r,y);
66 title('Output sequence using Overlap save method y[n
      ]');
67 xlabel('Samples n');
68 ylabel('Amplitude');
69
70 //INPUT:
71 //Enter the input sequence=[1 2 -1 2 3 -2 -3 -1 1 1
      2 -1]
72 //Enter the lower index of input sequence=0
73 //Enter the impulse response sequence=[1 2 3 -1]
74 //Enter the lower index of impulse response sequence
      =0
75
76 //OUTPUT:
77 //Output sequence using overlap save method Y(n):

```

```

78
79
80      //      column 1 to 12
81
82      //  1.      4.      6.      5.      2.      11.      0.  - 16.
           - 8.      3.      8.      5.
83
84      //      column 13 to 15
85
86      //  3.  - 5.      1.

```

Experiment: 12

WRITE A SCILAB PROGRAM TO FIND FFT USING DECIMATION IN TIME(DIT) METHOD.

Scilab code Solution 12.12 PROGRAM TO FIND FFT USING DECIMATION IN TIME METHOD

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 // CAPTION: PROGRAM TO FIND FFT USING DECIMATION IN
  TIME(DIT) METHOD
4 clc;
5 clear;
6 close;
7 x=input('Enter Input sequence='); // x=[0 1 2 3 4 5
    6 7] or x= [0 1 2 3]
8 N=length(x);
9
10 s=log2(N); // computing array size
11
12 //for sequence size 8
```

```

13 if (N==8) then
14 stage1=1;
15
16 x=[x(1) x(5) x(3) x(7) x(2) x(6) x(4) x(8)]; //
    stage1
17 for stage=1:s
18     for index=0:(2^stage):(N-1); //series of
        butterfly for each stage
19         for n=0:(stage1-1); // create butterfly
            and save result
20             pos=n+index+1; // index of data sample
21             pow=(2^(s-stage))*n; // part of power of
                complex multiplier
22             w=exp((-1*i)*(2*pi)*pow/N); // complex
                multiplier
23             a=x(pos)+x(pos+stage1).*w; // 1 st part
                of butterfly creating operations
24             b=x(pos)-x(pos+stage1).*w; // 2nd part of
                butterfly creating operation
25             x(pos)=a; // saving computation of 1st
                half
26             x(pos+stage1)=b; // saving computation
                of second part
27
28     end
29 end
30 stage1=2*stage1; // computing next stage
31 end
32
33 y=x;
34 disp('FFT of the given input sequence is y(n)=');
35 disp(y);
36
37 // for sequence size -4
38 else
39     stage1=1;
40 x=[x(1) x(3) x(2) x(4)]; // stage1
41 for stage=1:s

```

```

42     for index=0:(2^stage):(N-1); //series of
        butterfly for each stage
43         for n=0:(stage1-1); // create butterfly
            and save result
44             pos=n+index+1; // index of data sample
45             pow=(2^(s-stage))*n; // part of power of
                complex multiplier
46             w=exp((-1*i)*(2*pi)*pow/N); // complex
                multiplier
47             a=x(pos)+x(pos+stage1).*w; // 1 st part
                of butterfly creating operations
48             b=x(pos)-x(pos+stage1).*w; // 2nd part of
                butterfly creating operation
49             x(pos)=a; // saving computation of 1st
                half
50             x(pos+stage1)=b; // saving computation
                of second part
51
52     end
53 end
54 stage1=2*stage1; // computing next stage
55 end
56
57 y=x;
58 disp('FFT of the given input sequence y(n)=');
59 disp(y);
60 end
61
62 //INPUT:
63 //Enter Input sequence=[0 1 2 3 4 5 6 7]
64
65 //OUTPUT:
66 // FFT of the given input sequence is y(n)=
67
68
69 //      column 1 to 3
70
71 //      28.      - 4. + 9.6568542i      - 4. + 4.i

```

```

72
73      //      column 4 to 5
74
75 //      - 4. + 1.6568542 i      - 4.
76
77      //      column 6 to 7
78
79 //      - 4. - 1.6568542 i      - 4. - 4. i
80
81      //      column 8
82
83 //      - 4. - 9.6568542 i
84
85 // INPUT:
86 //Enter Input sequence=[0 1 2 3]
87
88 //OUTPUT:
89 //FFT of the given input sequence y(n)=
90
91 //      6.      - 2. + 2. i      - 2.      - 2. - 2. i

```

Experiment: 13

WRITE A SCILAB PROGRAM FOR DESIGNING OF FIR FILTER FOR LOW PASS, HIGH PASS, BANDPASS AND BAND REJECT RESPONSES.

Scilab code Solution 13.13 PROGRAM FOR DESIGNING OF FIR FILTER FOR LOW PASS HIGH PASS BAND PASS AND BAND REJECT RESPONSES

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 //CAPTION: PROGRAM FOR DESIGNING OF FIR LOW PASS,
  HIGH PASS, BAND PASS AND BAND REJECT RESPONSES
4
5 // Filter length=5 , Order of filter=4 , Window=
  Rectangular
```

```

6  clc;
7  clear;
8  close;
9  xdel(winsid());
10 fc1=input('Enter the analog cutoff Frequency in Hz='
    ); // 250
11 fc2=input('Enter analog higher cutoff frequency in
    Hz='); //600
12 fs=input('Enter the analog sampling Frequency in Hz=
    '); //2000
13 M=input('Enter order of filter='); //4
14 w1=(2*pi)*(fc1/fs);
15 w2=(2*pi)*(fc2/fs);
16
17 // FIR LOW PASS FILTER
18 disp('DESIGNING OF FIR LOW PASS FILTER:');
19 disp(w1, 'Digital Cutoff Frequency in radians.cycles
    /samples');
20 wc1=w1/pi;
21 disp(wc1, 'Normalized digital cutoff frequency in
    cycles/samples');
22 [wft,wfm,fr]=wfir('lp',M+1,[wc1/2,0], 're',[0,0]);
23 disp(wft, 'Impulse Response of LPF FIR filter:h(n)=')
    ;
24 //plotting magnitude response of LPF filter
25 subplot(2,4,1);
26 a=gca();
27 a.thickness=2;
28 a.foreground = 5;
29 a.font_color = 5;
30 a.font_style = 5;
31 plot(2*fr,wfm);
32 title('Magnitude Response of FIR LPF');
33 xlabel('Normalized Digital Frequency w');
34 ylabel('Magnitude |H(w)|');
35 xgrid(1)
36 subplot(2,4,2);
37 a=gca();

```



```

38 a.thickness=2;
39 a.foreground = 5;
40 a.font_color = 5;
41 a.font_style = 5;
42 plot(fr*fs,wfm);
43 title('Magnitude Response of FIR LPF');
44 xlabel('Analog frequency in Hz f');
45 ylabel('Magnitude |H(w)|');
46 xgrid(1)
47
48 //FIR HIGH PASS FILTER
49 disp('DESIGNING OF FIR HIGH PASS FILTER:')
50 disp(w1, 'Digital Cutoff Frequency in radians.cycles
    /samples');
51 wc1=w1/%pi;
52 disp(wc1,'Normalized digital cutoff frequency in
    cycles/samples');
53 [wft,wfm,fr]=wfir('hp',M+1,[wc1/2,0],'re',[0,0]);
54 disp(wft,'Impulse Response of HPF FIR filter:h(n)=')
    ;
55 //plotting magnitude response of HPF filter
56 subplot(2,4,3);
57 a=gca();
58 a.thickness=2;
59 a.foreground = 5;
60 a.font_color = 5;
61 a.font_style = 5;
62 plot(2*fr,wfm);
63 title('Magnitude Response of FIR HPF');
64 xlabel('Normalized Digital Frequency w');
65 ylabel('Magnitude |H(w)|');
66 xgrid(1)
67 subplot(2,4,4);
68 a=gca();
69 a.thickness=2;
70 a.foreground = 5;
71 a.font_color = 5;
72 a.font_style = 5;

```

```

73 plot(fr*fs,wfm);
74 title('Magnitude Response of FIR HPF');
75 xlabel('Analog frequency in Hz f');
76 ylabel('Magnitude |H(w)|');
77 xgrid(1)
78
79 // FIR BAND PASS FILTER
80 disp('DESIGNING OF FIR BAND PASS FILTER:');
81 disp(w1,'Digital lower cutoff frequency in radians
    cycles/samples');
82 disp(w2,'Digital higher cutoff frequency in radians
    cycles/samples');
83 wc1=w1/%pi;
84 wc2=w2/%pi;
85 disp(wc1,'Normalized digital lower cutoff frequency
    in cycles/samples');
86 disp(wc2,'Normalized digital higher cutoff frequency
    in cycles/samples');
87 [wft,wfm,fr]=wfir('bp', M+1,[wc1/2,wc2/2],'re',
    ,[0,0]);
88 disp(wft,'Impulse response of Bandpass Filter FIR
    filter:h(n)=');
89 //plotting the magnitude Response of HPF FIR filter
90 subplot(2,4,5);
91 a=gca();
92 a.thickness=2;
93 a.foreground = 5;
94 a.font_color = 5;
95 a.font_style = 5;
96 plot(2*fr, wfm);
97 xlabel('Normalized Digital Frequency w');
98 ylabel('Magnitude |H(W)|');
99 title('Magnitude Response of FIR BPF');
100 xgrid(1);
101 subplot(2,4,6);
102 a=gca();
103 a.thickness=2;
104 a.foreground = 5;

```

```

105 a.font_color = 5;
106 a.font_style = 5;
107 plot(fr*fs,wfm);
108 xlabel('Analog Frequency in Hz f');
109 ylabel('Magnitude |H(w)| ');
110 title('Magnitude response of FIR BPF');
111 xgrid(1);
112
113 // FIR BAND REJECT FILTER
114 disp('DESIGNING OF FIR BAND REJECT FILTER: ')
115 disp(w1,'Digital lower cutoff frequency in radians
        cycles/samples');
116 disp(w2,'Digital higher cutoff frequency in radians
        cycles/samples');
117 wc1=w1/%pi;
118 wc2=w2/%pi;
119 disp(wc1,'Normalized digital lower cutoff frequency
        in cycles/samples');
120 disp(wc2,'Normalized digital higher cutoff frequency
        in cycles/samples');
121 [wft,wfm,fr]=wfir('sb', M+1,[wc1/2,wc2/2],'re'
        ,[0,0]);
122 disp(wft,'Impulse response of Bandreject Filter FIR
        filter:h(n)=');
123 //plotting the magnitude Response of HPF FIR filter
124 subplot(2,4,7);
125 a=gca();
126 a.thickness=2;
127 a.foreground = 5;
128 a.font_color = 5;
129 a.font_style = 5;
130 plot(2*fr, wfm);
131 xlabel('Normalized Digital Frequency w');
132 ylabel('Magnitude |H(W)| ');
133 title('Magnitude Response of FIR BRF');
134 xgrid(1);
135 subplot(2,4,8);
136 a=gca();

```

```

137 a.thickness=2;
138 a.foreground = 5;
139 a.font_color = 5;
140 a.font_style = 5;
141 plot(fr*fs,wfm);
142 xlabel('Analog Frequency in Hz f');
143 ylabel('Magnitude |H(w)|');
144 title('Magnitude response of FIR BRF');
145 xgrid(1);
146
147 //INPUT:
148 //Enter the analog cutoff Frequency in Hz=250
149 //Enter analog higher cutoff frequency in Hz=600
150 //Enter the analog sampling Frequency in Hz=2000
151 //Enter order of filter=4
152
153 //OUTPUT:
154 // DESIGNING OF FIR LOW PASS FILTER:
155
156 // Digital Cutoff Frequency in radians.cycles/
    samples
157
158 //    0.7853982
159
160 // Normalized digital cutoff frequency in cycles/
    samples
161
162 //    0.25
163
164 //Impulse Response of LPF FIR filter:h(n)=
165
166 //    0.1591549    0.2250791    0.25    0.2250791
        0.1591549
167
168 //DESIGNING OF FIR HIGH PASS FILTER:
169
170 // Digital Cutoff Frequency in radians.cycles/
    samples

```

```

171
172 // 0.7853982
173
174 //Normalized digital cutoff frequency in cycles/
    samples
175
176 // 0.25
177
178 //Impulse Response of HPF FIR filter:h(n)=
179
180 // - 0.1591549 - 0.2250791 0.75 - 0.2250791 -
    0.1591549
181
182 //DESIGNING OF FIR BAND PASS FILTER:
183
184 //Digital lower cutoff frequency in radians cycles/
    samples
185
186 // 0.7853982
187
188 //Digital higher cutoff frequency in radians cycles
    /samples
189
190 // 1.8849556
191
192 //Normalized digital lower cutoff frequency in
    cycles/samples
193
194 // 0.25
195
196 //Normalized digital higher cutoff frequency in
    cycles/samples
197
198 // 0.6
199
200 //Impulse response of Bandpass Filter FIR filter:h(
    n)=
201

```

```

202    // - 0.2527039    0.0776516    0.35    0.0776516 -
          0.2527039
203
204    // DESIGNING OF FIR BAND REJECT FILTER:
205
206    // Digital lower cutoff frequency in radians cycles/
          samples
207
208    // 0.7853982
209
210    // Digital higher cutoff frequency in radians cycles
          /samples
211
212    // 1.8849556
213
214    // Normalized digital lower cutoff frequency in
          cycles/samples
215
216    // 0.25
217
218    // Normalized digital higher cutoff frequency in
          cycles/samples
219
220    // 0.6
221
222    // Impulse response of Bandreject Filter FIR filter :
          h(n)=
223
224    // 0.2527039 - 0.0776516    0.65 - 0.0776516
          0.2527039

```

Experiment: 14

WRITE A SCILAB PROGRAM TO DESIGN DIGITAL IIR BUTTERWORTH LOW PASS FILTER.

Scilab code Solution 14.14 PROGRAM TO DESIGN DIGITAL IIR BUTTERWORTH LOW PASS FILTER

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 //CAPTION: PROGRAM TO DESIGN BUTTERWORTH LOW PASS
  FILTER
4 clc;
5 clear;
6 close;
7 xdel(winsid());
8 fc=input('Enter cutoff freq in Hz fc='); //1000
9 fs=input('Enter sampling freq in Hz fs='); //10000
```

```

10 N=input('Enter order of Butterworth filter N=');  

11 Fp=2*fc/fs;  

12 [Hz]=iir(N,'lp','butt',[Fp/2,0],[0,0]);  

13 [Hw,w]=firmag(Hz,256);  

14 subplot(2,1,1);  

15 a=gca();  

16 a.thickness=2;  

17 a.foreground = 5;  

18 a.font_color = 5;  

19 a.font_style = 5;  

20 plot(2*w,abs(Hw));  

21 title('Magnitude Response of IIR LPF');  

22 xlabel('Normalized Digital frequency w');  

23 ylabel('Magnitude |H(w)|');  

24 xgrid(1);  

25 subplot(2,1,2);  

26 a=gca();  

27 a.thickness=2;  

28 a.foreground = 5;  

29 a.font_color = 5;  

30 a.font_style = 5;  

31 plot(2*w*fs,abs(Hw));  

32 title('Magnitude Response of IIR LPF');  

33 xlabel('Analog Frequency in Hz f');  

34 ylabel('Magnitude |H(w)|');  

35 xgrid(1);  

36  

37 //INPUT:  

38 //Enter cutoff freq in Hz fc=1000  

39 //Enter sampling freq in Hz fs=10000  

40 //Enter order of Butterworth filter N=4

```

Experiment: 15

WRITE A SCILAB PROGRAM TO DESIGN DIGITAL IIR CHEBYSHEW FILTER.

Scilab code Solution 15.15 PROGRAM TO DESIGN DIGITAL IIR CHEBYSHEW FILTER

```
1 // VERSION: Scilab: 5.4.1
2 // OS: windows 7
3 //CAPTION: PROGRAM TO DESIGN DIGITAL CHEBYSHEV IIR
  FILTER
4 clc;
5 clear;
6 close;
7 wp=input('Enter the Digital Pass Band edge Frequency
   ='); //0.2*%pi
8 ws=input('Enter the Digital Stop Band edge Frequency
   ='); //0.6*%pi
9 t=input('Sampling Interval='); //1
```

```

10 del1=input('Enter the Pass Band Ripple='); //0.8
11 del2=input('Enter the Stop Band Ripple='); //0.2
12 disp(wp,'Wp=');
13 disp(ws,'Ws=');
14 del=sqrt(((1/del2)^2)-1);
15 disp(del,'Delta=');
16 epsilon=sqrt(((1/del1)^2)-1);
17 disp(epsilon,'Epsilon=');
18 N=(acosh(del/epsilon))/(acosh(ws/wp));
19 N=ceil(N);
20 disp(N,'N=');
21 wc=wp/((((1/del1)^2)-1)^(1/(2*N)));
22 [pols,gn]=zpchl(N,epsilon,wp);
23 hs=poly(gn,'s','coeff')/real(poly(pols,'s'));
24 z=poly(0,'z');
25 hz=horner(hs,((2/t)*((z-1)/(z+1))));
26 hw=frmag(hz(2),hz(3),512); // freq. response for 512
    points
27 w=0:%pi/511:%pi;
28 a=gca();
29 a.thickness=2;
30 a.foreground = 5;
31 a.font_color = 5;
32 a.font_style = 5;
33 plot(w/%pi,abs(hw));
34 xgrid(1);
35 title('Magnitude Response of Digital Chebyshev LPF
    IIR Filter');
36 xlabel('Normalized digital Frequency');
37 ylabel('Magnitude in db');
38
39 //INPUT:
40 //Enter the Digital Pass Band edge Frequency=0.2*%pi
41 //Enter the Digital Stop Band edge Frequency=0.6*%pi
42 //Sampling Interval=1
43 //Enter the Pass Band Ripple=0.8
44 //Enter the Stop Band Ripple=0.2
45

```

```
46 //OUTPUT:
47 //Wp=
48
49 // 0.6283185
50
51 //Ws=
52
53 // 1.8849556
54
55 //Delta=
56
57 // 4.8989795
58
59 // Epsilon=
60
61 // 0.75
62
63 //N=
64
65 2.
```
