Scilab Manual for
Basic Simulation Laboratory
by Dr Kantipudi Mvv Prasad
Others
Sreyas Institute Of Engineering & Technology

Solutions provided by
Dr Kantipudi Mvv Prasad
Others
Sreyas Institute Of Engineering & Technology

September 22, 2020

\(^1\text{Funded 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 Basic operation on matrices 7

2 Generation of Various Signals & Sequences (Periodic/Aperiodic), such as Unit Impulse, Unit Step, Square, Sawtooth, Triangular 11

3 Write a program to perform operations on various Signals and Sequences 22

4 Finding the Even and Odd parts of Signal/Sequence and Real and Imaginary parts of Signal. 33

5 Convolution for Signals and sequences 41

6 Auto Correlation and Cross Correlation for Signals and Sequences 45

7 Verification of Linearity and Time Invariance Properties of a given Continuous/Discrete System 52

8 Computation of Unit sample, Unit step and Sinusoidal responses of the given LTI system and verifying its physical realizability 61

9 Gibbs Phenomenon Simulation 64
Finding the Fourier Transform of a given signal and plotting its magnitude and phase spectrum
<table>
<thead>
<tr>
<th>Solution</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Basic operations on matrices</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>Generation Of Unit Impulse and Unit Step Signal and Sequences</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Generation Of Square Wave and Sawtooth Wave Signals and Sequences</td>
<td>13</td>
</tr>
<tr>
<td>2.3</td>
<td>Generation Of Triangular and Sinusoidal Signal and Sequences</td>
<td>16</td>
</tr>
<tr>
<td>2.4</td>
<td>Generation Of Ramp and Sinc Signals and Sequences</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>Operations on Various Signals and Sequences</td>
<td>22</td>
</tr>
<tr>
<td>3.2</td>
<td>To perform Energy and Average Power Operations on Various Signals and Sequences</td>
<td>30</td>
</tr>
<tr>
<td>4.1</td>
<td>Finding Even and Odd Parts of the Signal</td>
<td>33</td>
</tr>
<tr>
<td>4.2</td>
<td>Finding Even and Odd Parts of Sequence</td>
<td>35</td>
</tr>
<tr>
<td>4.3</td>
<td>Finding Real and Imaginary parts of Signal or Sequence</td>
<td>37</td>
</tr>
<tr>
<td>5.1</td>
<td>Convolution of any two signals and sequences</td>
<td>41</td>
</tr>
<tr>
<td>6.1</td>
<td>Auto correlation of signals and sequences</td>
<td>45</td>
</tr>
<tr>
<td>6.2</td>
<td>Cross correlation of signals and sequences</td>
<td>47</td>
</tr>
<tr>
<td>7.1</td>
<td>Verifying linearity property of a given discrete system</td>
<td>52</td>
</tr>
<tr>
<td>7.2</td>
<td>Verifying the Time Invariance Property of a given Discrete System</td>
<td>57</td>
</tr>
<tr>
<td>8.1</td>
<td>Verifying Stability of a given LTI System</td>
<td>61</td>
</tr>
<tr>
<td>9.1</td>
<td>Verifying the Gibbs phenomenon</td>
<td>64</td>
</tr>
<tr>
<td>10.1</td>
<td>To find the Fourier Transform of a given signal and plotting its magnitude and phase spectrum</td>
<td>66</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Generation Of Unit Impulse and Unit Step Signal and Sequences</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>Generation Of Square Wave and Sawtooth Wave Signals and Sequences</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>Generation Of Triangular and Sinusoidal Signal and Sequences</td>
<td>17</td>
</tr>
<tr>
<td>2.4</td>
<td>Generation Of Ramp and Sinc Signals and Sequences</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>Operations on Various Signals and Sequences</td>
<td>23</td>
</tr>
<tr>
<td>3.2</td>
<td>Operations on Various Signals and Sequences</td>
<td>23</td>
</tr>
<tr>
<td>3.3</td>
<td>Operations on Various Signals and Sequences</td>
<td>24</td>
</tr>
<tr>
<td>4.1</td>
<td>Finding Even and Odd Parts of the Signal</td>
<td>35</td>
</tr>
<tr>
<td>4.2</td>
<td>Finding Even and Odd Parts of Sequence</td>
<td>37</td>
</tr>
<tr>
<td>4.3</td>
<td>Finding Real and Imaginary parts of Signal or Sequence</td>
<td>38</td>
</tr>
<tr>
<td>4.4</td>
<td>Finding Real and Imaginary parts of Signal or Sequence</td>
<td>38</td>
</tr>
<tr>
<td>5.1</td>
<td>Convolution of any two signals and sequences</td>
<td>44</td>
</tr>
<tr>
<td>5.2</td>
<td>Convolution of any two signals and sequences</td>
<td>44</td>
</tr>
<tr>
<td>6.1</td>
<td>Auto correlation of signals and sequences</td>
<td>47</td>
</tr>
<tr>
<td>6.2</td>
<td>Auto correlation of signals and sequences</td>
<td>48</td>
</tr>
<tr>
<td>6.3</td>
<td>Cross correlation of signals and sequences</td>
<td>50</td>
</tr>
<tr>
<td>6.4</td>
<td>Cross correlation of signals and sequences</td>
<td>51</td>
</tr>
<tr>
<td>7.1</td>
<td>Verifying the Time Invariance Property of a given Discrete System</td>
<td>56</td>
</tr>
<tr>
<td>7.2</td>
<td>Verifying the Time Invariance Property of a given Discrete System</td>
<td>56</td>
</tr>
<tr>
<td>8.1</td>
<td>Verifying Stability of a given LTI System</td>
<td>62</td>
</tr>
</tbody>
</table>
9.1 Verifying the Gibbs phenomenon .......................... 65

10.1 To find the Fourier Transform of a given signal and plotting its magnitude and phase spectrum .......................... 67
10.2 To find the Fourier Transform of a given signal and plotting its magnitude and phase spectrum .......................... 67
Experiment: 1

Basic operation on matrices

Scilab code Solution 1.1 Basic operations on matrices

```plaintext
// Experiment Number: 1
// Write a program to perform basic operation on matrices
// Basic Simulation Laboratory
// B. Tech II Year I Sem
// Student Name: Enrolement Number:
// Course Instructor: Dr. Kantipudi MVV Prasad,
// Sreyas Institute Of Engineering & Technology, Hyderabad.
//
// OS: Windows 10.1
// Scilab 6.0.2

clc;
close

```

7
// Enter Matrices from Keyboard
A = input('Enter the Matrix A :');
B = input('Enter the Matrix B :');

// Display the Entered Matrices from Keyboard
disp(A,'The Matrix A is .... : ');
disp(B,'The Matrix B is .... : ');

// Find the size of matrices
disp('The size of Matrix A is .... : ');
disp(size(A));
disp('The size of Matrix B is .... : ');
disp(size(B));

// Addition of two matrices
disp('Addition of A and B Matrices is .....: ');
disp(A + B);

// Subtraction of two matrices
disp('Subtraction of A and B Matrices is .....: ');
disp(A - B);

// Multiplication by a scalar
disp('Multiplication of matrix A with a scalar value K .....: ')
K = input('Enter a scalar value K :');
disp(K*A);

// Multiplication of two matrices

disp('Multiplication of A and B Matrices is ......: ');
disp(A * B);

// Multiplication (Element by Element) of two matrices

disp('Multiplication (Element by Element) of A and B Matrices is ......: ');
disp(A .* B);

// Finding the Rank of the matrix

disp('Rank of Matrix A is :');
disp(rank(A));

// Find the determinant of the matrix

disp('Determinant of Matrix A is :');
disp(det(A));

// Find the trace of the matrix

disp('Trace of Matrix A is :');
disp(trace(A));
// Find the Inverse of the matrix

disp('Inverse of Matrix A is :');

disp(inv(A));
Experiment: 2

Generation of Various Signals & Sequences (Periodic/Aperiodic), such as Unit Impulse, Unit Step, Square, Sawtooth, Triangular

Scilab code Solution 2.1  Generation Of Unit Impulse and Unit Step Signal and Sequences

1  //Experiment Number:2.1
2  //Write a program to generate unit impulse and unit step Signals and Sequences
3  //Basic Simulation Laboratory
4  //B.Tech II Year I Sem
5  //Student Name: Enrolment Number:
6  // Course Instructor: Dr. Kantipudi MV Prasad,
7  // Sreyas Institute Of Engineering & Technology,
8  // Hyderabad.
9  //
Figure 2.1: Generation Of Unit Impulse and Unit Step Signal and Sequences

```matlab
clc;
close;
clear;

// Unit Impulse Signal and Sequence

t=-4:1:4;
a=[zeros(1,4) 1 zeros(1,4)];
k=input("Enter the Amplitude :"); // reading amplitude value from keyboard
b=k*a;

subplot(2,2,1);
plot(t,b);
xlabel("Time");
ylabel("Amplitude");
```
title("Impulse Response");

subplot(2,2,2);
plot2d3(t,b);
xlabel("Time");
ylabel("Amplitude");
title("Impulse Response");

// Unit Step Signal and Sequence:

// Discrete Signal

t=0:3;
y=ones(1,4);

subplot(2,2,3);
plot2d3(t,y);
xlabel('Time');
ylabel('Amplitude');
title('Unit Step Discrete Signal');

// Continuous Signal

subplot(2,2,4);
plot(t,y);
xlabel('Time');
ylabel('Amplitude');
title('Unit Step Continuous Signal');

// Enter the Amplitude : 8

Scilab code Solution 2.2 Generation Of Square Wave and Sawtooth Wave Signals and Sequences
Figure 2.2: Generation Of Square Wave and Sawtooth Wave Signals and Sequences

1 // Experiment Number: 2.2
2 // Write a program to generate square wave and sawtooth wave Signals and Sequences
3 // Basic Simulation Laboratory
4 // B.Tech II Year I Sem
5 // Student Name: Enrollement Number:
6 // Course Instructor: Dr. Kantipudi MVV Prasad,
7 // Sreyas Institute Of Engineering & Technology, Hyderabad.
8 //
9
10 // OS : Windows 10.1
11 // Scilab 6.0.2
12
13 clc;
14 close;
15 clear ;
16
17
18
// continuous square wave Signal:

a=input('Enter Amplitude : ');
t=0:0.001:1;
d=a*squarewave(2*pi*10*t);

subplot(2,2,1);
plot(t,d);
xlabel("---------->Time Index t (sec.)");
ylabel("---------->Amplitude");
title(" Square Wave Signal ");

// discrete square wave signal

//a=input('Enter amplitude ');
n=0 : 0.01 :1;
d=a*squarewave(2*pi*10*n);

subplot(2,2,2);
plot2d3(n,d);
xlabel("---------->Time Index n");
ylabel("---------->Amplitude");
title("Square Wave Signal Sequence");

// Sawtooth Wave Signal

Fs = 20; // samples per second
 t_total = 10; // seconds
 n_samples = Fs * t_total;
 t = linspace(0, t_total, n_samples);
 f=500; // sound frequency

 saw_wave=2*(f*t-floor(0.5+f*t));

subplot(2,2,3);
plot(t,saw_wave);
xlabel("---------->Time Index t (sec.)");
ylabel("----------->Amplitude");
title ("Sawtooth Wave Signal");

// sawtooth wave sequence

Fs = 20; // samples per second
t_total = 10; // seconds
n_samples = Fs * t_total;
n = linspace (0, t_total, n_samples);
f = 500; // sound frequency

saw_wave = 2*(f*n-floor(0.5+f*n));

subplot (2,2,4);
plot2d3 (n, saw_wave);
xlabel ("--------->Time Index");
ylabel ("--------->Amplitude");
title ("Sawtooth Wave Signal Sequence");

// Input Parameters
// Enter Amplitude: 7

Scilab code Solution 2.3 Generation Of Triangular and Sinusoidal Signal and Sequences

// Experiment Number: 2.3
// Write a program to generate Triangular and Sinusoidal Signals and Sequences
// Basic Simulation Laboratory
// B.Tech II Year I Sem
// Student Name: Enrolement Number:
// Course Instructor: Dr. Kantipudi MVV Prasad,
Figure 2.3: Generation Of Triangular and Sinusoidal Signal and Sequences

```matlab
clc; 
close; 
clear; 

// Triangular Wave Signal 
Fs = 20; // samples per second 
t_total = 100; // seconds 
n_samples = Fs * t_total; 
t = linspace(0, t_total, n_samples); 
f=40; // sound frequency 
```
27  tri_wave=(2/\pi)*asin(sin(2*\pi*f*t));
28
29  subplot(2,2,1);
30
31  plot(t,tri_wave);
32  xlabel('--------->Time Index t (sec.)');
33  ylabel('---------->Amplitude');
34  title('Triangular Wave Signal ');
35
36  // triangular wave sequence
37
38  Fs = 20; // samples per second
39  t_total = 10; // seconds
40  n_samples = Fs * t_total;
41  n = linspace(0, t_total, n_samples);
42  f=40; // sound frequency
43
44  tri_wave=(2/\pi)*asin(sin(2*\pi*f*n));
45
46  subplot(2,2,2);
47  plot2d3(n,tri_wave);
48  xlabel('--------->Time Index t (sec.)');
49  ylabel('---------->Amplitude');
50  title('Triangular Wave Sequence ');
51
52  // continuous Sinusoidal Signal
53
54  a=input('Enter amplitude for Sinusoidal Signal: ');
55  t=0:0.001:1;
56  p=a*sin(2*\pi*10*t);
57
58  subplot(2,2,3);
59  plot(t,p);
60  title('Sinusoidal Signal');
61  xlabel('time');
62  ylabel('amplitude');
// discrete sinusoidal signal

//a=input('Enter magnitude');

n = 0:100;

x = a* sin(((2*0.05)*%pi)*n);

subplot(2,2,4);
plot2d3(n,x);
title("Sinusoidal Sequence");
xlabel("samples");
ylabel("magnitude");

// After Getting Traingular wave output, visit the command window to enter Input Parameters

// Enter amplitude for Sinusoidal Signal: 5

Scilab code Solution 2.4 Generation Of Ramp and Sinc Signals and Sequences
// Experiment Number: 2.4
// Write a program to generate ramp and sinc Signals and Sequences
// Basic Simulation Laboratory
// B.Tech II Year I Sem
// Studdent Name: Enrollment Number:
// Course Instructor: Dr. Kantipudi MVV Prasad,
// Sreyas Institute Of Engineering & Technology, Hyderabad.

// OS: Windows 10.1
// Scilab 6.0.2
clc;
close
clear;

// continuous ramp signal

t = 0 : 0.001 : 1;
y = 0.5 * t;

subplot(2,2,1);
plot(t, y);
xlabel ('Time Index t (sec.)');
ylabel ('Amplitude');
title ('Ramp Signal');

// discrete ramp signal

n = 0 : 0.1 : 1;
y = 0.5 * n;

subplot(2,2,2);
35 plot2d3(n,y);
36 xlabel ('----------->Time Index n');
37 ylabel ('----------->Amplitude');
38 title ('Ramp Signal Sequence');
39
40 //continuous sinc signal
41
t=linspace(-10 , 10);
y=sinc(t);
44 subplot(2,2,3);
45 plot(t,y);
46 xlabel("Time Index t (sec.)");
47 ylabel("Amplitude");
48 title("Sinc Signal");
50
51 //discrete sinc signal
52
53 n=linspace(-10 , 10);
54 y=sinc(n);
55
56 subplot(2,2,4);
57 plot2d3(n,y);
58 xlabel("Time Index n");
59 ylabel("Amplitude");
60 title("Sinc Signal Sequence");
Experiment: 3

Write a program to perform operations on various Signals and Sequences

Scilab code Solution 3.1 Operations on Various Signals and Sequences

```
1   //Experiment Number:  3.1
2   //Write a program to perform Addition, Multiplication
3       , Folding, Scaling and shifting operations on
4       various Signals and Sequences
5   //Basic Simulation Laboratory
6   //B.Tech II Year I Sem
7   //Student Name: ___________________________ Enrolement Number:
8   // Course Instructor: Dr. Kantipudi MVV Prasad,
9   // Sreyas Institute Of Engineering & Technology,
10  Hyderabad.
11  //
```
Figure 3.1: Operations on Various Signals and Sequences

Figure 3.2: Operations on Various Signals and Sequences
Figure 3.3: Operations on Various Signals and Sequences

```matlab
clc;
close all;

// Addition

disp('Enter the input sequences to perform Addition Operation');

x=input('Enter the sequence A=');
y=input('Enter the sequence B=');

m=length(x);
n=length(y);
```
if m==n then
  z=x+y;
  disp(z,'Addition result of two equal length sequences:=');
elseif m>n then
  y=[y,zeros(1,m-n)];
  z=x+y;
  disp(z,'Addition result of two unequal length sequences:=');
else
  x=[x,zeros(1,n-m)];
  z=x+y;
  disp(z,'Addition result of two unequal length sequences:=');
end

// Multiplication

disp('Enter the input sequences to perform multiplication Operation');

x=input('Enter the sequence A=');
y=input('Enter the sequence B=');

m=length(x);
n=length(y);
if \( m = n \) then
\[
z = x.*y;
\]
disp \((z, ' Multiplication result of two equal 
lengths sequences := ')

elseif \( m > n \) then
\[
y = [y, \text{zeros}(1, m-n)];
\]
z = x.*y;
disp \((z, ' Multiplication result of two unequal 
lengths sequences := ')

else
\[
x = [x, \text{zeros}(1, n-m)];
\]
z = x.*y;
disp \((z, ' Multiplication result of two unequal 
lengths sequences := ')

end

// Folding Operation

disp('Enter the input sequence to perform Folding 
Operation ');

x1 = input (' Enter the input sequence A := ');
m = length (x1);
s = input (' Enter the starting point of original 
signal= ');
h = s + m -1;
n = s :1: h;

subplot (2 ,1 ,1)
plot(n,x1)
plot2d3 (n,x1)
xlabel ( ' n===' )
ylabel ( ' Amplitude ' )
title ( ' Original Sequence ' )

subplot (2 ,1 ,2)
disp (n);
disp (-n);
plot (-n,x1)
plot2d3 (-n,x1)
xlabel ( ' n===' )
ylabel ( ' Amplitude ' )
title ( ' Folded Sequence ' )

// Scaling Operation

disp ('Enter the input sequence to perform Scaling Operation');

x2 = input ( ' Enter input Sequence := ' );
m = length(x2);
s= input ( ' Enter starting point of original signal := ' )
h = s+m-1;
n = s :1: h;
C = input ( 'Enter Compression Time Scaling factor: = ' )
n = s/C:1/C:h/C ;
disp(n);
figure;

subplot (2 ,1 ,1)
plot(x2)
plot2d3(x2)
xlabel(' n===')
ylabel(' Amplitude')
title(' Original Sequence ')

subplot(2,1,2)
plot(n,x2)
plot2d3(n,x2)
xlabel(' n===')
ylabel(' Amplitude')
title(' Time Scaling − Compressed Sequence ')

// shifting operation

disp('Enter the input sequence to perform shifting Operation');

x3 = input(' Enter the input sequence := ')
m = length(x3);
lx = input(' Enter the starting point of original signal := ')
hx = lx+m-1;

n = lx:1: hx;

l = input(' Enter the delay := ')

figure;

subplot(2,1,1)
plot(n,x3)
plot2d3(n,x3);
xlabel(' n===')
ylabel(' Amplitude')
title(' Original Sequence ')

n = lx+d:1: hx+d;
167 subplot (2,1,2)
168 disp(n);
169 plot(n,x3)
170 plot2d3(n,x3)
171 xlabel (’n——>’)
172 ylabel (’Amplitude’)
173 title (’Delayed Sequence ’)
174
175
176 // Enter the input sequences to perform Addition Operation
177 // Enter the sequence A=[1 3 5 7 9]
178 // Enter the sequence B=[1 3 5 ]
179 // Addition result of two unequal length sequences:=
180 // 2. 6. 10. 7. 9.
181 // Enter the input sequences to perform multiplication Operation
182 // Enter the sequence A=[1 3 5 7 9]
183 // Enter the sequence B=[1 3 5 ]
184 // Multiplication result of two unequal lengths sequences:=
185 // 1. 9. 25. 0. 0.
186
187 // Enter the input sequence to perform Folding Operation
188 // Enter the input sequence A := [1 3 5 ]
189
190 // Enter the starting point of original signal=1
191 // 1. 2. 3.
192 // -1. -2. -3.
193
194 // Enter the input sequence to perform Scaling Operation
195 // Enter input Sequence := [1 3 5 7 9]
196 // Enter starting point of original signal:= 1
197 // Enter Compression Time Scaling factor: = 0.5
Scilab code Solution 3.2 To perform Energy and Average Power Operations on Various Signals and Sequences

```matlab
clc;
close;
clear ;
```
// Energy and Average power of the given sequence

p = input('Enter the sequence = ');  
M = length(p);  
disp('The length of the Entered sequence is =');  
disp(M)  
sum = 0;  
for i = 1:M,  
    sum = sum + (i*i);  
end;  
disp('Energy of the given sequence is = ');  
Energy = sum  
disp(Energy);  
disp('Average Power of the given sequence is = ');  
Average_power = sum / M  
disp(Average_power)  

// Energy and Average power of a signal

t = 0:0.01:4;  
s = cos(2*%pi*t);  
M = length(s);  
disp('The length of the Entered Signal is =');  
disp(M)  
sum = 0;  
for i = 1:M,  
    sum = sum + (i*i);  
end;  
disp('Energy of the given signal is = ');  
Energy = sum  
disp(Energy)  
disp('Average Power of the given signal is = ');  
Average_power = sum / M  
disp(Average_power)
// Enter the sequence = [1 3 5 7 9]

// The length of the Entered sequence is = 5.

// Energy of the given sequence is = 55.

// Average Power of the given sequence is = 11.

// The length of the Entered Signal is = 401.

// Energy of the given signal is = 21574201.

// Average Power of the given signal is = 53801.
Experiment: 4

Finding the Even and Odd parts of Signal/Sequence and Real and Imaginary parts of Signal.

Scilab code Solution 4.1 Finding Even and Odd Parts of the Signal

1 // Experiment Number: 4.1
2 // Write a program to find Even and odd parts of the signal
3 // Basic Simulation Laboratory
4 // B.Tech II Year I Sem
5 // Student Name: Enrollement Number:
6 // Course Instructor: Dr. Kantipudi MV Prasad,
7 // Sreyas Institute Of Engineering & Technology,
   Hyderabad.
8 //
9
10 // OS : Windows 10.1
11 // Scilab 6.0.2
clc;
close
clear ;

// Even and odd parts of a signal

t=0:.005:4*pi;
x=sin(t)+cos(t); // x(t)=sint(t)+cos(t)

subplot(2,2,1)
plot2d3(t,x)
xlabel('t');
ylabel('amplitude')
title('input Signal f(t)')

y=sin(-t)+cos(-t) // y=x(-t)
subplot(2,2,2)
plot2d3(t,y)
xlabel('t');
ylabel('Amplitude')
title('Input Signal f(t)=−t')

z=x+y
subplot(2,2,3)
plot2d3(t,z/2)
xlabel('t');
ylabel('Amplitude')
title('Even Part of the signal')

p=x-y;
subplot(2,2,4)
plot2d3(t,p/2)
xlabel('t');
ylabel('Amplitude')
title('Odd Part of the signal');
Scilab code Solution 4.2 Finding Even and Odd Parts of Sequence

1 // Experiment Number: 4.2
2 // Write a program to find Even and odd parts of sequence
3 // Basic Simulation Laboratory
4 // B. Tech II Year I Sem
5 // Student Name: Enrolement Number:
6 // Course Instructor: Dr. Kantipudi MVV Prasad,
7 // Sreyas Institute Of Engineering & Technology, Hyderabad.
8 //
9 // OS : Windows 10.1
10 // Scilab 6.0.2
clc;
close;
clear;

// Even and Odd part of Sequence:
x = input('Enter the sequence : ');
y = -x;

subplot(2,2,1);
plot2d3(x);
xlabel('Time ---> '); ylabel('Amplitude --->');
title('Original signal f(t)');

subplot(2,2,2);
plot2d3(y);
xlabel('Time ---> '); ylabel('Amplitude --->');
title('Original signal f(-t)');

even = 0.5*(x + y);

subplot(2,2,3);
plot(even);
xlabel('Time ---> '); ylabel('Amplitude --->');
title('Even part');

odd = 0.5*(x - y);

subplot(2,2,4);
plot2d3(odd);
xlabel('Time ---> ');
Figure 4.2: Finding Even and Odd Parts of Sequence

```plaintext
50 ylabel('Amplitude ———>');
51 title('Odd part');
52
53 // Enter the sequence : [1 3 2 1]
```

**Scilab code Solution 4.3** Finding Real and Imaginary parts of Signal or Sequence

```plaintext
1 // Experiment Number: 4.3
2 // Write a program to find real and imaginary parts of signal/Sequence
3 // Basic Simulation Laboratory
4 // B.Tech II Year I Sem
5 // Student Name: Enrollement Number:
6 // Course Instructor: Dr. Kantipudi MVV Prasad,
```
Figure 4.3: Finding Real and Imaginary parts of Signal or Sequence

Figure 4.4: Finding Real and Imaginary parts of Signal or Sequence
clc; close clear;

// Real and Imaginary parts of even and odd signal:

x = input('Enter the complex Numbers Sequence: ');
n = 3:3;
xc = conj(x);
xc_folded = xc(:, $: -1 : 1);
xc_even = 0.5 *[x + xc_folded];
xc_odd = 0.5 *[x - xc_folded];

subplot(2,1,1);
plot2d3(n, real(xc_even));
title('Real part of even signal xc(n)')
xlabel ('n');
ylabel ('Magnitude of Real (xc-even)');

subplot(2,1,2);
plot2d3(n, imag(xc_even));
title('Imaginary part of even signal xc(n)')
xlabel ('n');
ylabel ('Magnitude of Imag (xc-even)');
figure;
subplot(2,1,1);
plot2d3(n, real(xc_odd));
title('Real part of odd signal xc(n)')
xlabel('n');
ylabel('Magnitude of Real (xc−odd)');

subplot(2,1,2);
plot2d3(n, imag(xc_odd));
title('Imaginary part of odd signal xc(n)')
xlabel('n');
ylabel('Magnitude of Imag (xc−odd)');

// Enter the complex Numbers Sequence: [3, 2+3*i, -3+2*i, 4-1*i, -2-3*i, 1-2*i, 1]
Experiment: 5

Convolution for Signals and sequences.

Scilab code Solution 5.1 Convolution of any two signals and sequences

```
1 //Experiment Number: 5
2 //Write a program to perform convolution of any two
3     signals and sequences.
4 //Basic Simulation Laboratory
5 //B.Tech II Year I Sem
6 //Student Name: Enrolement Number:
7 // Course Instructor: Dr. Kantipudi MVV Prasad ,
8 // Sreyas Institute Of Engineering & Technology ,
9     Hyderabad.
10 //
```

```
9
10 // OS : Windows 10.1
11 // Scilab 6.0.2
12
13 clc;
14 close
```
clear ;

// Convolution of two Sequences

x = input ('Enter the Input Sequence : ');
h = input ('Enter the Impulse Sequence : ');

subplot (3,1,1);
plot2d3(x);
plot(x)
title ('Input Sequence ')
xlabel (' n ');
ylabel (' x(n) ');

subplot (3,1,2);
plot2d3(h);
plot(h);
title ('Impulse Sequence ')
xlabel (' n ');
ylabel (' h(n) ');

Y = conv(x,h);
disp ('Convolved output = ');
disp(Y);

subplot (3,1,3);
plot2d3(Y);
plot(Y);
title("Linear Convolution of two Sequences");
xlabel (' n ');
ylabel (' Y(n) ');

// Convolution of two Signals

t = 1:20;
x = sin(t);
h = squarewave(t);
figure();
subplot(3,1,1);
plot2d3(x);
plot(x);
title('Input Signal')
xlabel ('n');
ylabel ('x(n)');

subplot(3,1,2);
plot2d3(h);
plot(h);
title('Impulse Response')
xlabel ('n');
ylabel ('h(n)');
o = conv(x,h);

subplot(3,1,3);
plot2d3(o);
plot(o);
title("Convolution of two Signals");
xlabel ('n');
ylabel ('Y(n)');

// Input Parameters
// Enter the Input Sequence : [1 2 3]
// Enter the Impulse Sequence : [-1 2 2]
// Convoluted output =
// -1. 0. 3. 10. 6.
Figure 5.1: Convolution of any two signals and sequences

Figure 5.2: Convolution of any two signals and sequences
Experiment: 6

Auto Correlation and Cross Correlation for Signals and Sequences.

Scilab code Solution 6.1 Auto correlation of signals and sequences

```plaintext
// Experiment Number: 6.1
// Write a program to compute Auto correlation and Cross correlation between signals and sequences.
// Basic Simulation Laboratory
// B. Tech II Year I Sem
// Student Name: Enrolment Number:
// Course Instructor: Dr. Kantipudi MVV Prasad
// Sreyas Institute Of Engineering & Technology, Hyderabad.

//
```

// OS: Windows 10.1
// Scilab 6.0.2

45
clc;
close
clear;

// Auto correlation of a sequence

a = input('Enter the sequence ....:: :');
res = xcorr(a);
disp(res);

subplot(2,1,1);
plot2d3(a);
xlabel('——> Samples');
ylabel('——> Amplitude');
title('Input Sequence');

subplot(2,1,2);
plot2d3(res);
xlabel('——> Samples');
ylabel('——> Amplitude');
title('Output Sequence');

// Auto correlation of a signal

t = 0:0.01:2;
a = cos(2 * %pi * t);
res = xcorr(a);
figure();

subplot(2,1,1);
plot(a);
xlabel('——> Samples');
ylabel('——> Amplitude');
Figure 6.1: Auto correlation of signals and sequences

```matlab
52 title('Input Sequence');
53 subplot(2,1,2);
54 plot(res);
55 xlabel('--- Samples');
56 ylabel('--- Amplitude');
57 title('Output Sequence');
58
60 // Input Arguments
61 // Enter the sequence . . . : [ 1 2 5 7]
62 // 7. 19. 47. 79. 47. 19. 7.
```

**Scilab code Solution 6.2** Cross correlation of signals and sequences

```matlab
1 //Experiment Number: 6.2
```
// Write a program to compute Cross correlation between signals and sequences.
// Basic Simulation Laboratory
// B. Tech II Year I Sem
// Student Name: Enrolment Number:
// Course Instructor: Dr. Kantipudi MVV Prasad,
// Sreyas Institute Of Engineering & Technology,
// Hyderabad.

// OS : Windows 10.1
// Scilab 6.0.2

clc;
close
clear;

// Cross correlation of a two sequences
a = input('Enter the first sequence ..::: ');
b = input('Enter the second sequence . . .: ');

r = xcorr(a,b);

subplot(3,1,1);
plot2d(a);
xlabel('------> Samples');
ylabel('------> Amplitude');
title('Input Sequence (1)');

subplot(3,1,2);
plot2d(b);
xlabel('------> Samples');
ylabel('------> Amplitude');
title('Input Sequence (2)');

subplot(3,1,3);
plot2d(r);
xlabel('------> Samples');
ylabel('------> Amplitude');
title('Cross correlation of a two sequences');

// Cross correlation of a two signals

$\mathbf{t} = 0:0.01:2;$
a = \cos(2 \times \pi \times t);
b = \sin(2 \times \pi \times t);
res = xcorr(a,b);
figure();

subplot(3,1,1);
plot(a);
xlabel('------> Samples');
ylabel('------> Amplitude');
title('Input signal (1)');

subplot(3,1,2);
Figure 6.3: Cross correlation of signals and sequences

59 plot(b);
60 xlabel(’−−−−> Samples’);
61 ylabel(’−−−−> Amplitude’);
62 title(’Input Signal(2)’);
63 subplot(3,1,3);
64 plot(res);
65 xlabel(’−−−−> Samples’);
66 ylabel(’−−−−> Amplitude’);
67 title(’Cross correlation of a two signals’);
68
69 // Enter the first sequence ....:: [ 1 2 3 7]
70
71 // Enter the second sequence ....:: [ 1 2 5 7]
Figure 6.4: Cross correlation of signals and sequences
Experiment: 7

Verification of Linearity and Time Invariance Properties of a given Continuous/Discrete System

Scilab code Solution 7.1 Verifying linearity property of a given discrete system

1  //Experiment Number: 7.1
2  //Write a program to Verify linearity property of a given continuous/discrete system.
3  //Basic Simulation Laboratory
4  //B.Tech II Year I Sem
5  //Studdent Name: Enrolement Number:
6  //Course Instructor: Dr. Kantipudi MVV Prasad,
7  //Sreyas Institute Of Engineering & Technology, Hyderabad.
8  //

9
10

52
clc; clear all; close;

// Enter two input sequences and impulse sequence

x1 = input (' Enter the samples of x1 ');
x2 = input (' Enter the samples of x2 ');
if(length(x1)~=length(x2))
disp('Error...!: Lengths of two input sequences [x1 and x2] are different'); return;
end;

h = input (' Enter the samples of h ');

// Length of the output sequence

N = length(x1) + length(h) -1;
disp('length of the output signal will be ');
disp(N);

// Enter scaling factors

a1 = input (' The scale factor a1 is ');
a2 = input (' The scale factor a2 is ');
x = a1 * x1 + a2 * x2;

// Response of x and x1
yo1 = conv(x,h);
y1 = conv(x1,h);

// scaled response of x1
y1s = a1 * y1;

// Response of x2
y2 = conv(x2,h);

// Scaled Response of x2
y2s = a2 * y2;

yo2 = y1s + y2s;

disp ('Input signal x1 is ');
disp(x1);

disp ('Input signal x2 is ');
disp(x2);

disp ('Output Sequence yo1 is ');
disp(yo1);

disp ('Output Sequence yo2 is');
disp(yo2);

if ( yo1 == yo2 )

disp(' yo1 = yo2. Hence the LTI system is LINEAR ')
end;

// Enter the samples of x1 [ 1 5 7 9]
// Enter the samples of x2 [ 4 3 2 2]

// Enter the samples of h [1 2 2 2]

// length of the output signal will be 7.

// The scale factor a1 is 2

// The scale factor a2 is 3

// Input signal x1 is

// 1. 5. 7. 9.

// Input signal x2 is

// 4. 3. 2. 2.

// Output Sequence yo1 is

// 14. 47. 86. 130. 126. 88. 48.

// Output Sequence yo2 is

// 14. 47. 86. 130. 126. 88. 48.

// yo1 = yo2. Hence the LTI system is LINEAR
Figure 7.1: Verifying the Time Invariance Property of a given Discrete System

Figure 7.2: Verifying the Time Invariance Property of a given Discrete System
Scilab code Solution 7.2 Verifying the Time Invariance Property of a given Discrete System

```scilab
// Experiment Number: 7.2
// Write a program to Verify the Time Invariance of a given Discrete System.
// Basic Simulation Laboratory
// B. Tech II Year I Sem
// Student Name: Enrollement Number:
// Course Instructor: Dr. Kantipudi MVV Prasad,
// Sreyas Institute Of Engineering & Technology,
// Hyderabad.
// OS : Windows 10.1
// Scilab 6.0.2
clc;
clear all;
close;
// Entering two input sequences
x = input( ' Enter the samples of signal x(n) ' );
h = input( ' Enter the samples of signal h(n) ' );
// original response
y = conv(x,h);
disp( ' Enter a POSITIVE number for delay ' );
d = input( ' Desired delay of the signal is ' );
//Delayed input
```

57
xd = [zeros(1,d), x];
nxd = 0 : length(xd)-1;

// Delayed output
yd = conv(xd, h);
nyd = 0 : length(yd)-1;

disp('Original Input Signal x(n) is ');
disp(x);
disp('Delayed Input Signal xd(n) is ');
disp(xd);
disp('Original Output Signal y(n) is ');
disp(y);
disp('Delayed Output Signal yd(n) is ');
disp(yd);

xp = [x, zeros(1,d)];
subplot(2,1,1);
plot2d3(nxd, xp);
plot(nxd, xp);
xgrid(2);
xlabel('Time Index n ');
ylabel('x(n) ');
title('Original Input Signal x(n) ');

subplot(2,1,2);
plot2d3(nxd, xd);
plot(nxd, xd);
xgrid(2);
xlabel('Time Index n ');
ylabel('xd(n) ');
title('Delayed Input Signal xd(n) ');
yp = [y zeros(1,d)];

figure;

subplot(2,1,1);
plot2d3(nyd,yp);
plot(nyd,yp);
xgrid(2)
xlabel(' Time Index n ');
ylabel(' y(n) ');
title(' Original Output Signal y(n) ');

subplot(2,1,2);
plot2d3(nyd,yd);
plot(nyd,yd);
xgrid(2)
xlabel(' Time Index n ');
ylabel(' yd(n) ');
title(' Delayed Output Signal yd(n) ');

// Enter the samples of signal x(n) [1 2 3 8 5 6]
// Enter the samples of signal h(n) [2 3 5 4 4 6]
// Enter a POSITIVE number for delay
// Desired delay of the signal is 2

// Original Input Signal x(n) is
// 1. 2. 3.  8. 5.  6.

// Delayed Input Signal xd(n) is
// 0. 0. 1. 2. 3. 8. 5. 6.

// Original Output Signal y(n) is

// 2. 7. 17. 39. 61. 93. 99. 100.
// 92. 54. 36.

// Delayed Output Signal yd(n) is

// 0. 0. 2. 7. 17. 39. 61. 93. 99.
// 100. 92. 54. 36.
Experiment: 8

Computation of Unit sample, Unit step and Sinusoidal responses of the given LTI system and verifying its physical realizability

Scilab code Solution 8.1 Verifying Stability of a given LTI System

1 //Experiment Number: 8
2 //Write a program to compute the Unit sample, unit step and sinusoidal response of the given LTI system and verifying its stability.
3 //Basic Simulation Laboratory
4 //B.Tech II Year I Sem
5 //Student Name: Enrollement Number:
6 // Course Instructor: Dr. Kantipudi MV Prasad,
7 // Sreyas Institute Of Engineering & Technology, Hyderabad.
8 //
Figure 8.1: Verifying Stability of a given LTI System

9 // OS : Windows 10.1
10 // Scilab 6.0.2
11 // Stability of a given LTI System:
12
13 clc;
14 clear all;
15 close;
16 n=0:0.1:20;
17
18 h= input("Enter the System Equation : "); // 0.2* sin (0.3*n).*cos(0.2*%pi*n)
19
20 sum=0;
21
22 for k=1:201
23 if abs(h(k))<10^-6
24 end
25 sum=sum+h(k);
end
disp('The summation value is . . . .: ');
disp(sum);

if sum > 5.0983e+008
disp('The System is unstable');
else
disp('The System is stable');
end;
plot2d3(n,h);
xgrid(2);
xlabel('n ==> ')
ylabel('h(k) ==> ')
title('Stability')

// Enter the System Equation : 0.2 * sin(0.3 * n) * cos
(0.2 * %pi * (n-1))

// The summation value is . . . .:

// 0.5909252

// The System is stable
Experiment: 9

Gibbs Phenomenon Simulation

Scilab code Solution 9.1 Verifying the Gibbs phenomenon

1
2 // Experiment Number: 9
3 // Write a program to verify the Gibbs phenomenon
4 // Basic Simulation Laboratory
5 // B. Tech II Year I Sem
6 // Student Name: Enrolement Number:
7 // Course Instructor: Dr. Kantipudi MVV Prasad,
8 // Sreyas Institute Of Engineering & Technology,
9 // Hyderabad.
10 //
11 // OS : Windows 10.1
12 // Scilab 6.0.2
13
c1c;
14 clear all;
15 close ;
16
64
Figure 9.1: Verifying the Gibbs phenomenon

```
17 J = 500   // number of points
18 x = linspace(0, 2*pi, J);
19 f = sign(x);   // returns array same size as x
20 kp = 0.*x;  // multiplies everything by x starting with 0
21 t = 150
22 for k = 1:2:t
23 kp = kp + (1/2) * sin(k*x)/k;
24 end
25 plot2d3(x, kp);
26 plot(x, kp);
27 xlabel('time');
28 ylabel('Amplitude');
29 title('Gibbs phenomenon');
```
Experiment: 10

Finding the Fourier Transform of a given signal and plotting its magnitude and phase spectrum

Scilab code Solution 10.1  To find the Fourier Transform of a given signal and plotting its magnitude and phase spectrum

1 // Experiment Number: 10
2 // Write a program to find the Fourier Transform of a given signal and plotting its magnitude and phase spectrum.
3 // Basic Simulation Laboratory
4 // B.Tech II Year I Sem
5 // Student Name: Enrollement Number:
6 // Course Instructor: Dr. Kantipudi MV Prasad,
7 // Sreyas Institute Of Engineering & Technology, Hyderabad.
Figure 10.1: To find the Fourier Transform of a given signal and plotting its magnitude and phase spectrum

Figure 10.2: To find the Fourier Transform of a given signal and plotting its magnitude and phase spectrum
clc;
clear all;
close;

f=150  //(' Frequency in hertz ');
Fs=2000  //('Sample freq in khz ')
Ts=1/(Fs);
N=128  //('DFT sequence ');
n=[0:N-1]*Ts;
x=0.8*cos(2*%pi*f*n);
plot(n,x);
set (gca () ," grid",[1 1]);
data_bounds =([0 -1 ; 0.05 1]) ;
title(' Cosine signal frequency ');
xlabel(' Time in n (sec) ');
ylabel(' Amplitude ');
Y=fft(x);
w=[0:N-1];
figure;
Xmag=abs(Y);
subplot(2,1,1);
plot(w,Xmag);
set (gca ()," grid",[1 1]);
title(' Magnitude of fourier transform ');
xlabel(' Frequency index w-------> ');
ylabel(' Magnitude -------> ');
Xphase= atan(imag(Y),real(Y));
subplot(2,1,2);
plot(w,Xphase);
44 set(gca(),"grid",[1 1]);
45 title('Phase of fourier transform ');
46 xlabel('Frequency index w\rightarrow ');
47 ylabel('Phase \rightarrow ');