

Scilab Manual for  
Signals & Systems  
by Prof Akhtar Nadaf  
Electronics and Telecommunication  
Engineering  
N.K. Orchid College Of Engineering &  
Technology, Solapur<sup>1</sup>

Solutions provided by  
Prof Akhtar Nadaf  
Electronics and Telecommunication Engineering  
N.k. Orchid College Of Engineering & Technology Solapur

October 17, 2021

<sup>1</sup>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	3
1 Generation of continuous & discrete time signals	5
2 Basic operations on CT & DT Signals	11
3 Determination of Even & odd part of signal	20
4 Determination of energy & power of signal	24
5 Sampling Theorem	26
6 Linear convolution sum and integral	29
7 Fourier series	33
8 Fourier transform	40
9 Laplace Transform	44
10 Z transform & Inverse Z transform	46
11 Response of LTI System	48
12 Probability	50
13 Auto correlation	52
14 Cross correlation	55

# List of Experiments

Solution 1.1	Generation of Continuous Time Signals . . . . .	5
Solution 1.2	Generation of Discrete Time signals . . . . .	7
Solution 2.1	Basic Operation on CT Signals . . . . .	11
Solution 2.2	Basic Operation on CT Signals . . . . .	13
Solution 2.3	Basic Operatiion on DT Signals . . . . .	15
Solution 2.4	Basic Operation on DT Signals . . . . .	17
Solution 3.1	Determination of even and odd part of CT signal	20
Solution 3.2	Determination of even and odd part of DT signal	22
Solution 4.1	Determination of energy and power of CT signal .	24
Solution 5.1	Sampling Theorem . . . . .	26
Solution 6.1	Linear convolution . . . . .	29
Solution 6.2	Convolution Integral . . . . .	30
Solution 7.1	Fourier Series of CT Signal . . . . .	33
Solution 7.2	Fourier Series of DT Signal . . . . .	35
Solution 8.1	Fourier Transform of CT Signal . . . . .	40
Solution 8.2	Discrete Time Fourier Transform . . . . .	42
Solution 9.1	Laplace transform . . . . .	44
Solution 10.1	Z transform and Inverse Z transform of given se- quence . . . . .	46
Solution 12.1	Estimation of probability for tossing of coin . . .	50
Solution 13.1	Auto Correlation of DT signal . . . . .	52
Solution 14.1	Cross correlation . . . . .	55

# List of Figures

1.1	Generation of Continuous Time Signals . . . . .	7
1.2	Generation of Discrete Time signals . . . . .	10
2.1	Basic Operation on CT Signals . . . . .	13
2.2	Basic Operation on CT Signals . . . . .	15
2.3	Basic Operatiion on DT Signals . . . . .	17
2.4	Basic Operation on DT Signals . . . . .	19
3.1	Determination of even and odd part of CT signal . . . . .	21
3.2	Determination of even and odd part of DT signal . . . . .	23
5.1	Sampling Theorem . . . . .	28
6.1	Linear convolution . . . . .	31
6.2	Convolution Integral . . . . .	32
7.1	Fourier Series of CT Signal . . . . .	36
7.2	Fourier Series of CT Signal . . . . .	37
7.3	Fourier Series of CT Signal . . . . .	37
7.4	Fourier Series of CT Signal . . . . .	38
7.5	Fourier Series of DT Signal . . . . .	39
8.1	Fourier Transform of CT Signal . . . . .	41
8.2	Discrete Time Fourier Transform . . . . .	43
11.1	Step Response of LTI System . . . . .	49
11.2	Step Response of LTI System . . . . .	49
13.1	Auto Correlation of DT signal . . . . .	54
14.1	Cross correlation . . . . .	56

# Experiment: 1

## Generation of continuous & discrete time signals

Scilab code Solution 1.1 Generation of Continuous Time Signals

```
1 // VERSION : Scilab : 6.0.1
2 // OS : WINDOWS 10
3 // CAPTION: GENERATION OF CONTINUOUS TIME SIGNALS
4 //UNIT IMPULSE SIGNAL
5 clc ;
6 clear all;
7 close ;
8 N =7; //SET LIMIT
9 t1 = -N:N;
10 x1=[zeros(1,N),1,zeros(1,N)];
11 subplot (2,3,1);
12 plot (t1,x1);
13 xgrid(4,1,7); //xgrid([color],[thickness],[style])
14 xlabel("Time");
15 ylabel("Amplitude");
16 title("Unit Impulse Signal");
17
18 //UNIT STEP SIGNAL
19 t2 =0:4;
```

```

20 x2= ones (1,5);
21 subplot (2,3,2);
22 plot (t2,x2);
23 xgrid(4,1,7);
24 xlabel ("time");
25 ylabel ("amplitude");
26 title ("Unit Step Signal");
27
28 //EXPONENTIAL SIGNAL
29 t3 =0:1:20;
30 x3=exp(-t3);
31 subplot (2,3,3);
32 plot (t3,x3);
33 xgrid(4,1,7);
34 xlabel ("time");
35 ylabel ("Amplitude");
36 title ("Exponential Signal");
37
38 //RAMP SIGNAL
39 t4 = -0:20;
40 x4=t4;
41 subplot (2,3,4);
42 plot (t4 ,x4);
43 xgrid(4,1,7);
44 xlabel ("Time");
45 ylabel ( " Amplitude");
46 title ("Ramp Signal");
47
48 //SINUSOIDAL SIGNAL
49
50 t5 =0:0.04:1;
51 x5=sin(2*%pi*t5);
52 subplot(2,3,5);
53 plot (t5,x5);
54 xgrid(4,1,7);
55 title ("Sinusoidal Signal")
56 xlabel ("Time");
57 ylabel ("Amplitude");

```

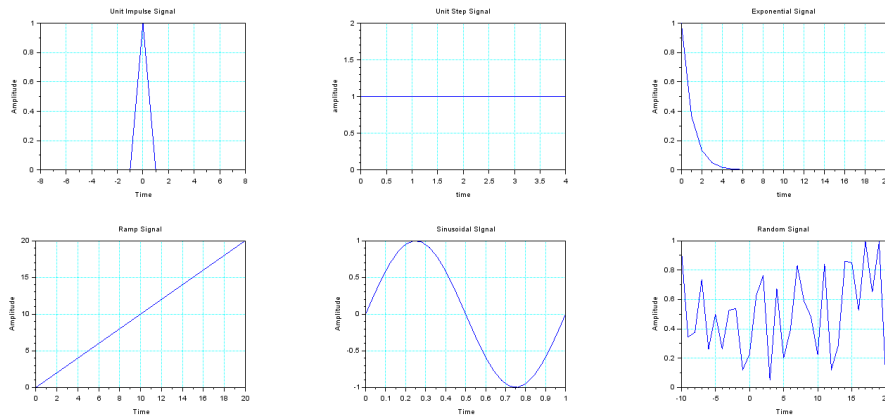


Figure 1.1: Generation of Continuous Time Signals

```

58
59 //RANDOM SIGNAL
60 t6=-10:1:20;
61 x6=rand(1,31);
62 subplot(2,3,6);
63 plot(t6,x6);
64 xgrid(4,1,7);
65 xlabel("Time");
66 ylabel("Amplitude");
67 title("Random Signal");

```

---

### Scilab code Solution 1.2 Generation of Discrete Time signals

```

1 //O.S. Windows 10
2 //Scilab 6.0.1
3 //Caption: Generation of Discrete time signals
4 clear;
5 clc;
6 // DT Unit step Signal

```



```

7 t =0:4;
8 y=ones(1,5);
9 subplot(3,2,1);
10 plot2d3(t,y);
11 xgrid(4,1,7); //xgrid([color],[thickness],[style])
12 xlabel("Time");
13 xlabel("n");
14 ylabel("Amplitude");
15 title("DT Unit Step Signal");
16
17 //DT Unit Ramp signal
18 n1 =0:8;
19 //Figure 1.1: Waveform generation using DT signals
20 y1=n1;
21 subplot(3,2,2);
22 plot2d3(n1,y1);
23 xgrid(4,1,7); //xgrid([color],[thickness],[style])
24 xlabel("Time");
25 xlabel("n");
26 ylabel("Amplitude");
27 title("DT Unit Ramp");
28
29 //DT Growing Exponential Signal
30 n1 =0:8;
31 y1=n1;
32 y2=exp(n1);
33 subplot(3,2,3);
34 plot2d3(n1,y2);
35 xgrid(4,1,7); //xgrid([color],[thickness],[style])
36 xlabel("Time");
37 xlabel("n");
38 ylabel("Amplitude");
39 title("DT Growing Exponential signal");
40
41 //DT Decaying exponential Signal
42 n1 =0:8;
43 y1=n1;
44 y2=exp(-n1);

```

```

45 subplot (3 ,2 ,4);
46 plot2d3 (n1 ,y2);
47 xgrid(4,1,7);
48 xlabel("n");
49 ylabel(" Amplitude");
50 title("DT Decaying Exponential Signal" );
51
52 //DT Sinusoidal signal
53 n1 =0:25;
54 y1=n1;
55 y2=sin(n1);
56 subplot (3 ,2 ,5);
57 plot2d3 (n1 ,y2);
58 xgrid(4,1,7);
59 xlabel("n");
60 ylabel(" Amplitude" );
61 title("DT Sinusoidal Signal");
62
63 //DT Unit Impulse signal
64 l=7;
65 n=-1:1;
66 x=[zeros(1,1),1,zeros(1,1)];
67 b=gca();
68 b.y_location =" middle";
69 subplot(3,2,6);
70 plot2d3( " gnn" ,n,x);
71 xgrid(4,1,7);
72 a= gce ();
73 a.children (1).thickness =5;
74 xtitle("DT Unit sample sequence");
75 xlabel("n");
76 ylabel(" Amplitude" );

```

---

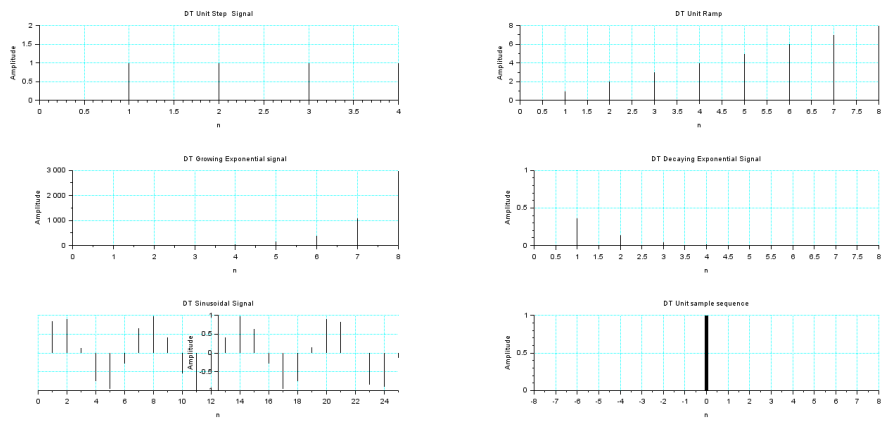


Figure 1.2: Generation of Discrete Time signals

# Experiment: 2

## Basic operations on CT & DT Signals

Scilab code Solution 2.1 Basic Operation on CT Signals

```
1 // VERSION : Scilab : 6.0.1
2 // OS : WINDOWS 10
3 // CAPTION: BASIC OPERATION ON CONTINUOUS TIME
  SIGNALS
4 clc ;
5 clear all;
6 close ;
7 //Input CT Signal
8 t = -2:1:2;
9 x = [1, 1, 0, 1, 0];
10 subplot (2, 3, 1);
11 plot (t, x);
12 xgrid (4, 1, 7);
13 xlabel ("Time");
14 ylabel ("Amplitude");
15 title ("Input CT Signal x(t)");
16
17 //Amplitude Scaling
18 a = 2; // Scaling factor
```

```

19 subplot (2,3,2);
20 plot(t,a*x);
21 xgrid(4,1,7);
22 xlabel("Time");
23 ylabel("Amplitude");
24 title("z(t)=2x(t)");
25
26 //Amplitude Scaling
27 a=0.5; // Scaling factor
28 subplot (2,3,3);
29 plot(t,a*x);
30 xgrid(4,1,7);
31 xlabel("Time");
32 ylabel("Amplitude");
33 title("z(t)=0.5x(t)");
34
35 // Time reversal
36 subplot (2,3,4);
37 plot(-t,x);
38 xgrid(4,1,7);
39 xlabel("Time");
40 ylabel("Amplitude");
41 title("z(t)=x(-t)");
42
43 //Time Shifting
44 subplot (2,3,5);
45 plot(t+2,x);
46 xgrid(4,1,7);
47 xlabel("Time");
48 ylabel("Amplitude");
49 title("z(t)=x(t+2)");
50
51 //Time Shifting
52 subplot (2,3,6);
53 plot(t-2,x);
54 xgrid(4,1,7);
55 xlabel("Time");
56 ylabel("Amplitude");

```

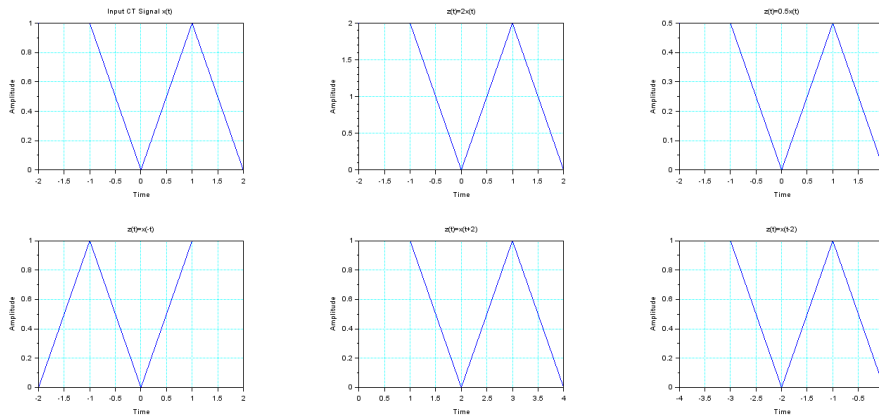


Figure 2.1: Basic Operation on CT Signals

57 `title("z(t)=x(t-2)");`

---

### Scilab code Solution 2.2 Basic Operation on CT Signals

```

1 // VERSION : Scilab : 6.0.1
2 // OS : WINDOWS 10
3 // CAPTION: BASIC OPERATION ON CONTINUOUS TIME
  SIGNALS
4 clc ;
5 clear all;
6 close ;
7 //Input CT Signal
8 t=-2:1:2;
9 x=[1,1,0,1,0];
10 subplot (2,3,1);
11 plot(t,x);
12 xgrid(4,1,7);
13 xlabel("Time");
14 ylabel("Amplitude");

```

```

15 title("Input CT Signal x(t)");
16
17 //Time Scaling
18 a=2; // Scaling factor
19 subplot (2,3,2);
20 plot(a*t,x);
21 xgrid(4,1,7);
22 xlabel("Time");
23 ylabel("Amplitude");
24 title("y(t)=x(2*t)");
25
26 //Time Scaling
27 a=0.5; // Scaling factor
28 subplot (2,3,3);
29 plot(a*t,x);
30 xgrid(4,1,7);
31 xlabel("Time");
32 ylabel("Amplitude");
33 title("y(t)=x(0.5*t)");
34
35 //Signal Addition
36 //Input Signal y(t)
37 t2=-2:1:2;
38 y=[1,1,0,1,1];
39 subplot (2,3,4);
40 plot(t2,y);
41 xgrid(4,1,7);
42 xlabel("Time");
43 ylabel("Amplitude");
44 title("Input signal y(t)");
45
46 //Signal addition
47 subplot (2,3,5);
48 plot(n,x+y);
49 xgrid(4,1,7);
50 xlabel("Time");
51 ylabel("Amplitude");
52 title("z(t)= x(t)+y(t)");

```

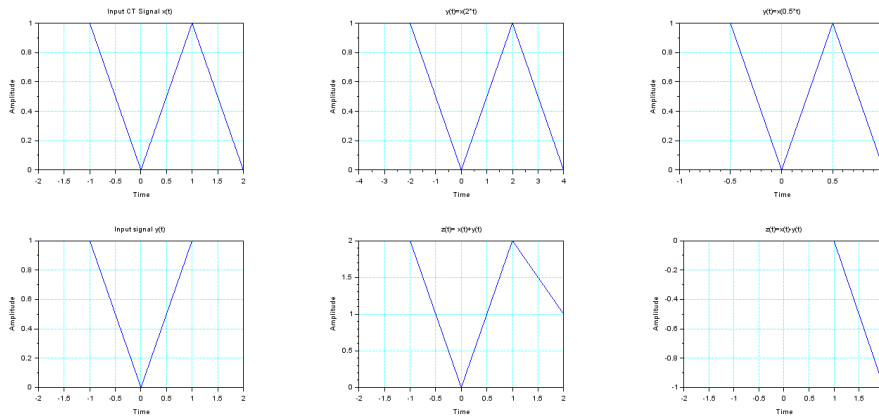


Figure 2.2: Basic Operation on CT Signals

```

53
54 //Signal subtraction
55 subplot (2,3,6);
56 plot(n,x-y);
57 xgrid(4,1,7);
58 xlabel("Time");
59 ylabel("Amplitude");
60 title("z(t)=x(t)-y(t)");

```

---

### Scilab code Solution 2.3 Basic Operatiion on DT Signals

```

1 // VERSION : Scilab : 6.0.1
2 // OS : WINDOWS 10
3 // CAPTION: BASIC OPERATION ON DISCRETE TIME SIGNALS
4 clc ;
5 clear all;
6 close ;
7 //Input DT Signal
8 n=-2:1:2;

```



```

 9 x=[1,1,0,0.5,0.5];
10 subplot (2,3,1);
11 plot2d3(n,x,style=[color("blue")]);
12 xgrid(4,1,7);
13 xlabel("n");
14 ylabel("Amplitude");
15 title("Input DT Signal x(n)");
16
17 //Amplitude Scaling
18 a=2; // Scaling factor
19 subplot (2,3,2);
20 plot2d3(n,a*x,style=[color("blue")]);
21 xgrid(4,1,7);
22 xlabel("n");
23 ylabel("Amplitude");
24 title("z(n)=2*x(n)");
25
26 //Amplitude Scaling
27 a=0.5; // Scaling factor
28 subplot (2,3,3);
29 plot2d3(n,a*x,style=[color("blue")]);
30 xgrid(4,1,7);
31 xlabel("n");
32 ylabel("Amplitude");
33 title("z(n)=0.5*x(n)");
34
35 // Time reversal
36 subplot (2,3,4);
37 plot2d3(-n,x,style=[color("blue")]);
38 xgrid(4,1,7);
39 xlabel("n");
40 ylabel("Amplitude");
41 title("z(n)=x(-n)");
42
43 //Time Shifting
44 subplot (2,3,5);
45 plot2d3(n+2,x,style=[color("blue")]);
46 xgrid(4,1,7);

```

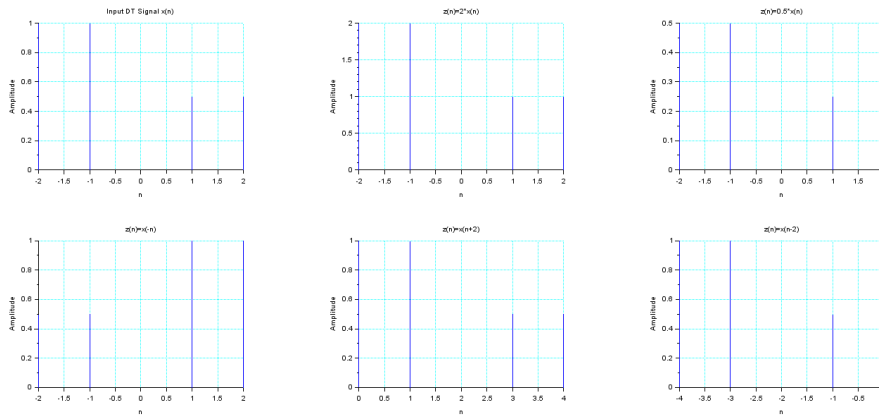


Figure 2.3: Basic Operation on DT Signals

```

47 xlabel("n");
48 ylabel("Amplitude");
49 title("z(n)=x(n+2)");
50
51 //Time Shifting
52 subplot (2,3,6);
53 plot2d3(n-2,x,style=[color("blue")]);
54 xgrid(4,1,7);
55 xlabel("n");
56 ylabel("Amplitude");
57 title("z(n)=x(n-2)");

```

---

### Scilab code Solution 2.4 Basic Operation on DT Signals

```

1 // VERSION : Scilab : 6.0.1
2 // OS : WINDOWS 10
3 // CAPTION: BASIC OPERATION ON DISCRETE TIME SIGNALS
4 clc ;
5 clear all;

```

```

6  close ;
7  //Input DT Signal
8  n=-2:1:2;
9  x=[1,1,0,0.5,0.5];
10 subplot (2,3,1);
11 plot2d3(n,x,style=[color("blue")]);
12 xgrid(4,1,7);
13 xlabel("n");
14 ylabel("Amplitude");
15 title("Input DT Signal x(n)");
16
17
18 //Time Scaling
19 a=2; // Scaling factor
20 subplot (2,3,2);
21 plot2d3(a*n,x,style=[color("blue")]);
22 xgrid(4,1,7);
23 xlabel("n");
24 ylabel("Amplitude");
25 title("z(n)=2*x(n)");
26
27 //Time Scaling
28 a=0.5; // Scaling factor
29 subplot (2,3,3);
30 plot2d3(a*n,x,style=[color("blue")]);
31 xgrid(4,1,7);
32 xlabel("n");
33 ylabel("Amplitude");
34 title("z(n)=0.5*x(n)");
35
36 //Signal Addition
37 //Input Signal y(n)
38 n2=-2:1:2;
39 y=[1,1,0,1,1];
40 subplot (2,3,4);
41 plot2d3(n2,y,style=[color("blue")]);
42 xgrid(4,1,7);
43 xlabel("n");

```

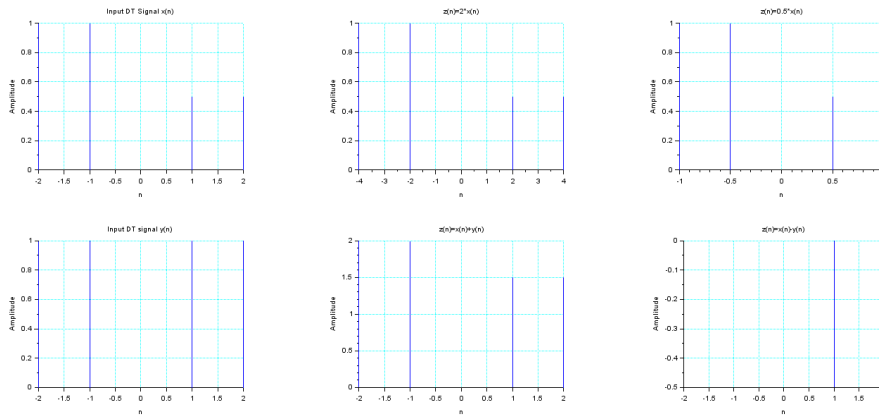


Figure 2.4: Basic Operation on DT Signals

```

44 ylabel(" Amplitude");
45 title("Input DT signal y(n)");
46
47 //Signal addition
48 subplot (2,3,5);
49 plot2d3(n,x+y,style=[color(" blue")]);
50 xgrid(4,1,7);
51 xlabel("n");
52 ylabel(" Amplitude");
53 title("z(n)=x(n)+y(n)");
54
55 //Signal subtraction
56 subplot (2,3,6);
57 plot2d3(n,x-y,style=[color(" blue")]);
58 xgrid(4,1,7);
59 xlabel("n");
60 ylabel(" Amplitude");
61 title("z(n)=x(n)-y(n)");

```

---

## Experiment: 3

# Determination of Even & odd part of signal

Scilab code Solution 3.1 Determination of even and odd part of CT signal

```
1 // VERSION : Scilab : 6.0.1
2 // OS : WINDOWS 10
3 // CAPTION: DETERMINATION OF EVEN & ODD PART OF CT
  SIGNAL
4 clc ;
5 clear all;
6 close ;
7 //Input CT Signal
8 t = -2 : 1 : 2;
9 x = [1, 1, 0, 1, 0];
10 subplot (2, 3, 1);
11 plot (t, x);
12 xgrid (4, 1, 7);
13 xlabel ("Time");
14 ylabel ("Amplitude");
15 title ("Input CT Signal x(t)");
16 c = 3;
17 for j = 1 : length (t)
```

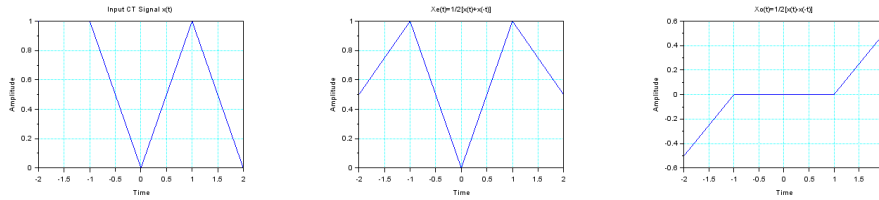


Figure 3.1: Determination of even and odd part of CT signal

```

18     i=n(j);
19     xe(j)=(1/2)*(x(c+i)+x(c-i));
20     xo(j)=(1/2)*(x(c+i)-x(c-i));
21 end
22 xe=[xe(c-2),xe(c-1),xe(c+0),xe(c+1),xe(c+2)];
23 xo=[xo(c-2),xo(c-1),xo(c+0),xo(c+1),xo(c+2)];
24 subplot(2,3,2);
25 plot(t,xe);
26 xgrid(4,1,7);
27 xlabel("Time");
28 ylabel("Amplitude");
29 title("Xe(t)=1/2[x(t)+x(-t)]");
30 subplot(2,3,3);
31 plot(t,-xo);
32 xgrid(4,1,7);
33 xlabel("Time");
34 ylabel("Amplitude");
35 title("Xo(t)=1/2[x(t)-x(-t)]");

```

---

**Scilab code Solution 3.2** Determination of even and odd part of DT signal

```
1 // VERSION : Scilab : 6.0.1
2 // OS : WINDOWS 10
3 // CAPTION: DETERMINATION OF EVEN & ODD PART OF DT
  SIGNAL
4 clc ;
5 clear all;
6 close ;
7 //Input DT Signal
8 n=-2:1:2;
9 x=[-2,1,2,-1,3];
10 subplot (2,3,1);
11 plot2d3(n,x,style=[color(" blue")]);
12 xgrid(4,1,7);
13 xlabel("n");
14 ylabel(" Amplitude");
15 title(" Input DT Signal x(t)");
16 c=3;
17 for j=1:length(n)
18     i=n(j);
19     xe(j)=(1/2)*(x(c+i)+x(c-i));
20     xo(j)=(1/2)*(x(c+i)-x(c-i));
21 end
22 xe=[xe(c-2),xe(c-1),xe(c+0),xe(c+1),xe(c+2)]; //
  calculation of even Part
23 xo=[xo(c-2),xo(c-1),xo(c+0),xo(c+1),xo(c+2)]; //
  ccalculation of odd part
24 subplot (2,3,2);
25 plot2d3(n,xe,style=[color(" blue")]);
26 xgrid(4,1,7);
27 xlabel("n");
28 ylabel(" Amplitude");
29 title("Xe(n)=1/2[x(n)+x(-n)]");
30 subplot (2,3,3);
31 plot2d3(n,-xo,style=[color(" blue")]);
32 xgrid(4,1,7);
```

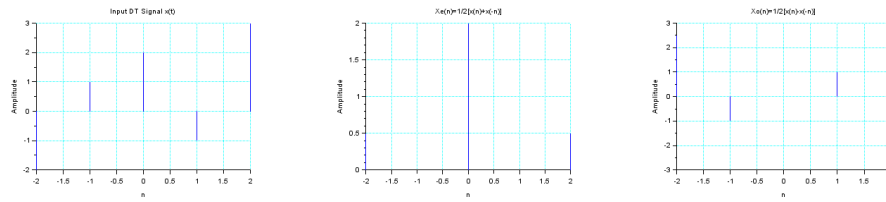


Figure 3.2: Determination of even and odd part of DT signal

```

33 xlabel("n");
34 ylabel("Amplitude");
35 title("Xo(n) = 1/2[x(n) - x(-n)]");

```

---



## Experiment: 4

# Determination of energy & power of signal

Scilab code Solution 4.1 Determination of energy and power of CT signal

```
1 // VERSION : Scilab : 6.0.1
2 // OS : WINDOWS 10
3 // CAPTION: DETERMINATION OF ENERGY & POWER OF SIGNAL
4 // Calculate energy of signal  $y(t)=5e^{-5t}$ 
5 clc ;
6 clear all;
7 t = 0:0.001:10;
8 y = 5*exp (-5*t);
9 E = integrate ( '(5*exp(-5*t))^2', 't', 0, 2*%pi ); //
    Calculation of energy of signal
10 disp (E, 'Energy of given signal in Joules=' ); //
    Expected output: 2.5 Joules
11 // Expected output in console
12 // Energy of given signal in Joules=
13 // 2.5
14 // Calculate power of  $x(t)=Ae^{-5t}$ 
15 t = 0:0.001:10;
16 y = 5*exp (-5*t);
17 P = integrate ( '(5*exp(-5*t))^2', 't', 0, 2*%pi ) / (2*%pi)
```

```
    ;  
18 disp(P, 'power of the signal in Watts=');  
19 //Expected Output in console  
20 // power of the signal in Watts=  
21 //    0.3978874
```

---

# Experiment: 5

## Sampling Theorem

Scilab code Solution 5.1 Sampling Theorem

```
1 // VERSION : Scilab : 6.0.1
2 // OS : WINDOWS 10
3 // CAPTION: Sampling Theorem
4 clc ;
5 clear all;
6 t =0:0.01:100;
7 f =0.02;
8 x=sin(2*%pi*f*t);
9 subplot(2,2,1);
10 plot (t,x);
11 xgrid(4,1,7);
12 xlabel("Time");
13 ylabel("Amplitude");
14 title ("CT Signal");
15 // Sampling frequency less than twices of input
    signal frequency
16 fs1 =0.002;
17 n =0:0.1:100;
18 x1=sin (2*%pi*f*n/fs1 );
19 subplot(2,2,2);
20 plot2d3(n,x1,style=[color("blue")]);
```

```
21 xlabel("n");
22 ylabel(" Amplitude");
23 xgrid(4,1,7);
24 title( "fs<2f");
25 // Sampling frequency is equal to twices of input
    signal frequency
26 fs2 =0.04;
27 x2= sin (2*%pi*f*n/fs2);
28 subplot(2,2,3);
29 plot2d3(n,x2,style=[color(" blue")]);
30 xlabel("n");
31 ylabel(" Amplitude");
32 xgrid(4,1,7);
33 title ("Sampling fs=2f");
34 //Sampling frequency is greater than twice of input
    signal frequency
35 fs3 =0.4;
36 x3= sin (2*%pi*f*n/fs3 );
37 subplot (2,2,4);
38 plot2d3 (n,x3,style=[color(" blue")]);
39 xlabel("n");
40 ylabel(" Amplitude");
41 xgrid(4,1,7);
42 title ("fs>2f");
```

---

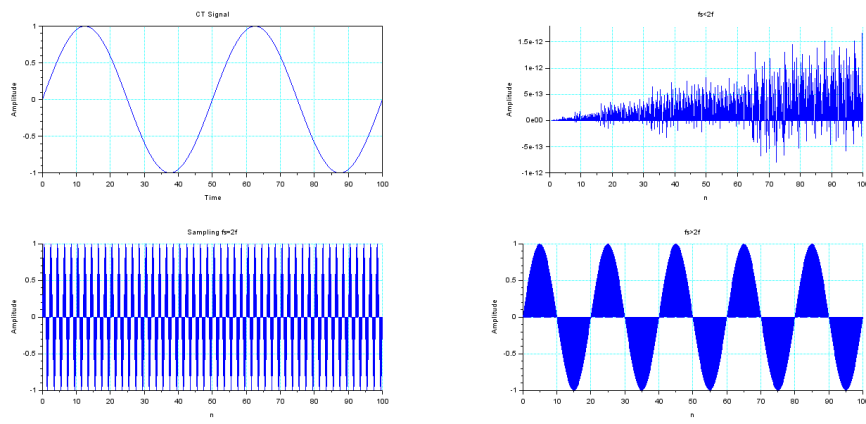


Figure 5.1: Sampling Theorem

# Experiment: 6

## Linear convolution sum and integral

Scilab code Solution 6.1 Linear convolution

```
1 // VERSION : Scilab : 6.0.1
2 // OS : WINDOWS 10
3 // CAPTION: Convolution sum of two sequences x(n)
   = [1,4,3,2] & h(n) = [1,3,2,1]
4 clc ;
5 clear all;
6 clc;
7 x=[1 4 3 2];
8 t1=0:1:3
9 subplot(2,3,1);
10 plot2d3(t1,x,style=[color("blue")]);
11 xgrid(4,1,7);
12 xlabel("n");
13 ylabel("Amplitude");
14 title("Input DT Signal x(n)");
15
16 h=[1 3 2 1];
17 t2=0:1:3
18 subplot(2,3,2);
```

```

19 plot2d3(t2,h,style=[color("blue")]);
20 xgrid(4,1,7);
21 xlabel("n");
22 ylabel("Amplitude");
23 title("Input DT Signal h(n)");
24 y=convol(x,h);
25 disp(y,"Convolution sum of the above two sequences
      is:");
26 // Expected output in console:
27 // Convolution sum of the above two sequences is:
28
29 //   1.   7.   17.   20.   16.   7.   2.
30
31 l=length(y);
32 t3=0:l-1;
33 subplot(2,3,3);
34 plot2d3(t3,y,style=[color("blue")]);
35 xgrid(4,1,7);
36 xlabel("n");
37 ylabel("Amplitude");
38 title("y(n)=x(n)*h(n)");

```

---

### Scilab code Solution 6.2 Convolution Integral

```

1 // Scilab Version: 6.0.1
2 //O.S : Windows 10
3 //Caption: Convolution integral of  $x(t)=(e^{-at}).u(t)$ 
      and  $h(t)=u(t)$ 
4 clear;
5 close;
6 clc;
7 T=10;
8 h=ones(1,T);

```

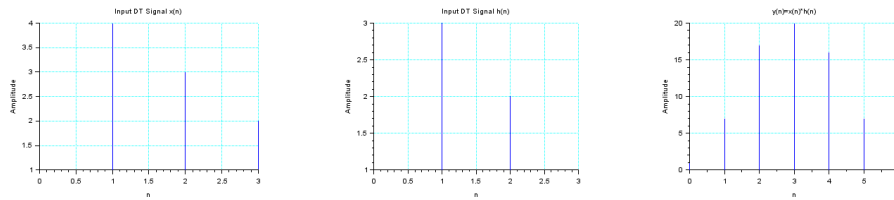


Figure 6.1: Linear convolution

```

9 N2=0:length(h)-1;
10 a=0.5;
11 for t=1:T
12     x(t)=exp(-a*(t-1));
13 end
14 N1=0:length(x)-1;
15 y=conv(x,h)-1;
16 N=0:length(x)+length(h)-2;
17 subplot(2,3,1);
18 plot(N1,x)
19 xlabel("t");
20 ylabel("Amplitude");
21 xgrid(4,1,7);
22 title("Input signal x(t)=(e^{-at}).u(t)")
23 subplot(2,3,2);
24 plot(N2,h);
25 xlabel("t");
26 ylabel("Amplitude");
27 xgrid(4,1,7);
28 title("Input signal h(t)=u(t)");
29 subplot(2,3,3);
30 plot(N(1:T),y(1:T))
31 xlabel("t");

```



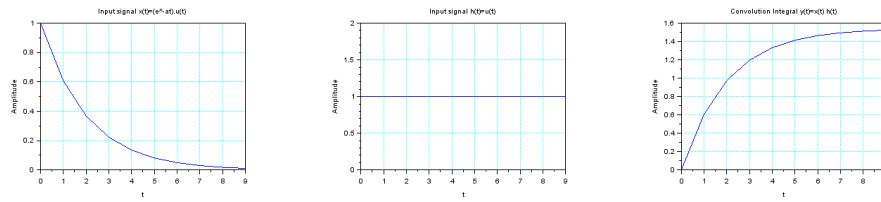


Figure 6.2: Convolution Integral

```

32 ylabel(" Amplitude");
33 title(" Convolution Integral y(t)=x(t) h(t)");
34 xgrid(4,1,7);

```

---

# Experiment: 7

## Fourier series

Scilab code Solution 7.1 Fourier Series of CT Signal

```
1 // Scilab Vrsion: 6.0.1
2 //O.S : Windows 10
3 //Caption: CTFS of a periodic signal x(t)
4 clear;
5 close;
6 clc;
7 t = -3:0.01:3;
8 //t1 = -%pi*4:(%pi*4)/100:%pi*4;
9 //t2 =-%pi*6:(%pi*6)/100:%pi*6;
10 xot = ones(1,length(t));
11 x1t = (1/2)*cos(%pi*2*t);
12 xot_x1t = xot+x1t;
13 x2t = cos(%pi*4*t);
14 xot_x1t_x2t = xot+x1t+x2t;
15 x3t = (2/3)*cos(%pi*6*t);
16 xt = xot+x1t+x2t+x3t;
17 //
18 figure
19 a = gca();
20 a.y_location = "origin";
21 a.x_location = "origin";
```

```

22 a.data_bounds=[-4,0;2 4];
23 plot(t,xot)
24 ylabel('t')
25 title('xot =1')
26 //
27 figure
28 subplot(2,1,1)
29 a = gca();
30 a.y_location = "origin";
31 a.x_location = "origin";
32 a.data_bounds=[-4,-3;2 4];
33 plot(t,x1t)
34 ylabel('t')
35 title('x1(t) =1/2*cos(2*pi*t)')
36 subplot(2,1,2)
37 a = gca();
38 a.y_location = "origin";
39 a.x_location = "origin";
40 a.data_bounds=[-4,0;2 4];
41 plot(t,xot_x1t)
42 ylabel('t')
43 title('xo(t)+x1(t)')
44 //
45 figure
46 subplot(2,1,1)
47 a = gca();
48 a.y_location = "origin";
49 a.x_location = "origin";
50 a.data_bounds=[-4,-2;4 2];
51 plot(t,x2t)
52 ylabel('t')
53 title('x2(t) =cos(4*pi*t)')
54 subplot(2,1,2)
55 a = gca();
56 a.y_location = "origin";
57 a.x_location = "origin";
58 a.data_bounds=[-4,0;4 4];
59 plot(t,xot_x1t_x2t)

```

```

60 ylabel('t')
61 title('x0(t)+x1(t)+x2(t)')
62 //
63 figure
64 subplot(2,1,1)
65 a = gca();
66 a.y_location = "origin";
67 a.x_location = "origin";
68 a.data_bounds=[-4,-3;4 3];
69 plot(t,x3t)
70 ylabel('t')
71 title('x1(t) =2/3*cos(6*pi*t)')
72 subplot(2,1,2)
73 a = gca();
74 a.y_location = "origin";
75 a.x_location = "origin";
76 a.data_bounds=[-4,-3;4 3];
77 plot(t,xt)
78 ylabel('t')
79 title('x(t)=x0(t)+x1(t)+x2(t)+x3(t)')

```

---

### Scilab code Solution 7.2 Fourier Series of DT Signal

```

1 // Scilab Version: 6.0.1
2 //O.S : Windows 10
3 //Caption: Discrete Time Fourier Transform of  $x[n]=\sin(\text{Won})$ 

```

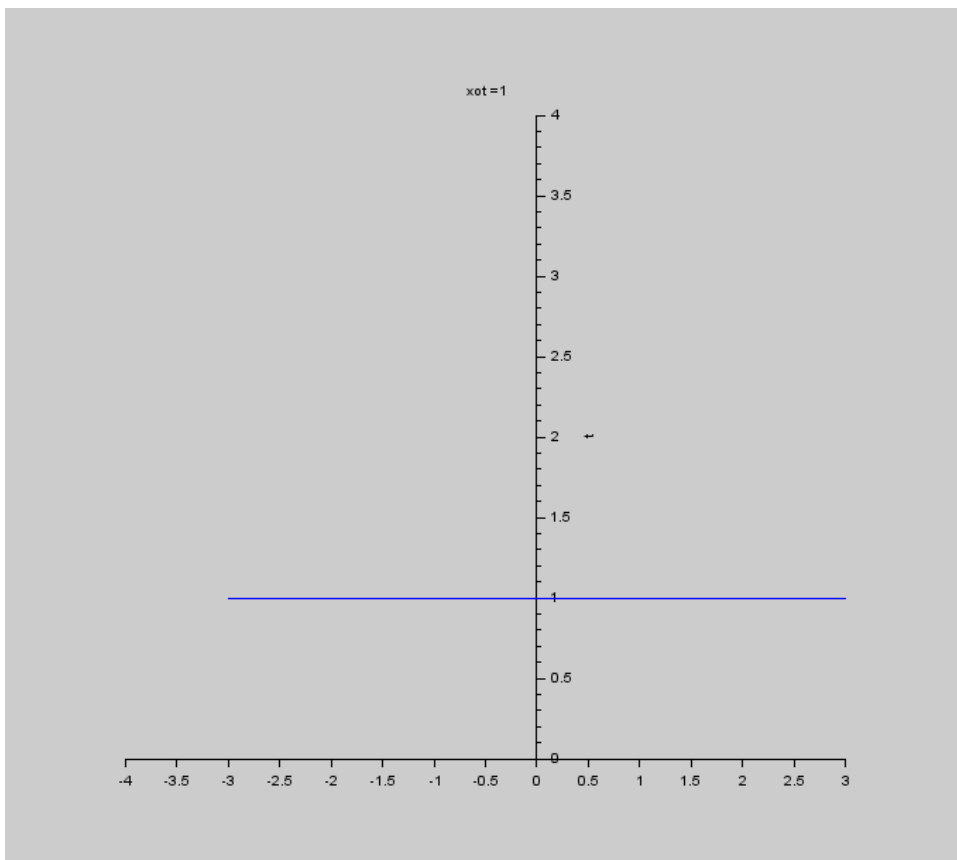


Figure 7.1: Fourier Series of CT Signal

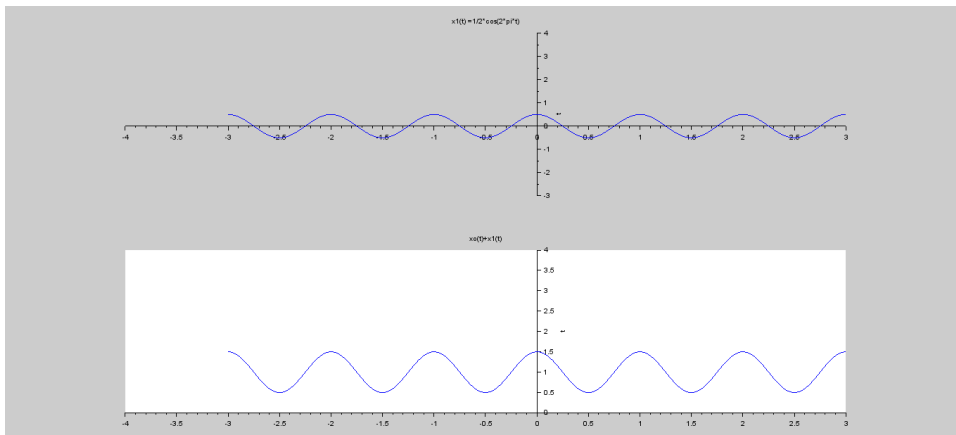


Figure 7.2: Fourier Series of CT Signal

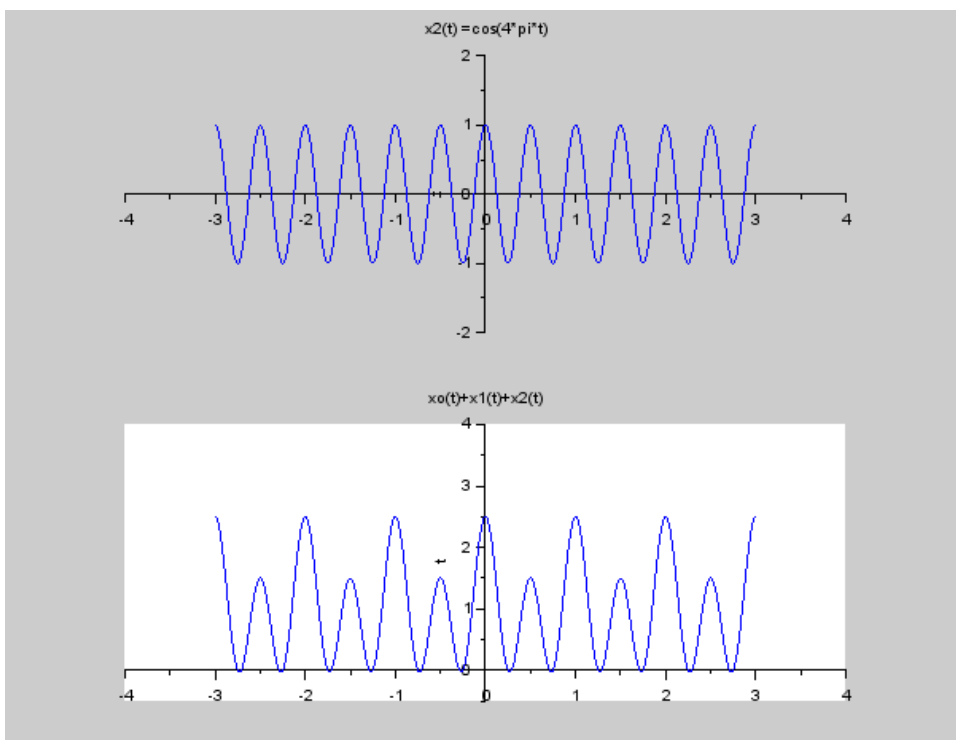


Figure 7.3: Fourier Series of CT Signal

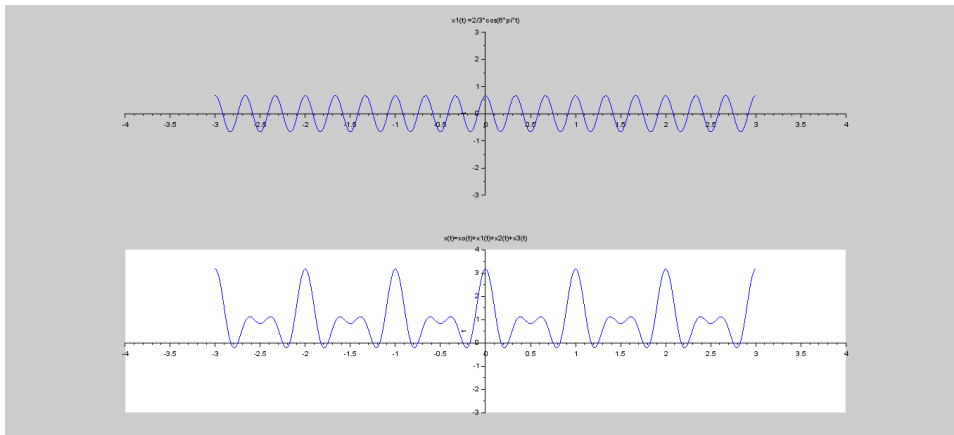


Figure 7.4: Fourier Series of CT Signal

```

4 clear;
5 close;
6 clc;
7 n = 0:0.01:5;
8 N = 5;
9 Wo = 2*%pi/N;
10 xn = sin(Wo*n);
11 for k = 0:N-2
12     C(k+1,:) = exp(-sqrt(-1)*Wo*n.*k);
13     a(k+1) = xn*C(k+1,:)'/length(n);
14     if(abs(a(k+1))<=0.01)
15         a(k+1)=0;
16     end
17 end
18 a = a'
19 a_conj = conj(a);
20 ak = [a_conj($:-1:1),a(2:$)]
21 k = -(N-2):(N-2);
22 //
23 figure
24 a = gca();
25 a.y_location = "origin";
26 a.x_location = "origin";

```

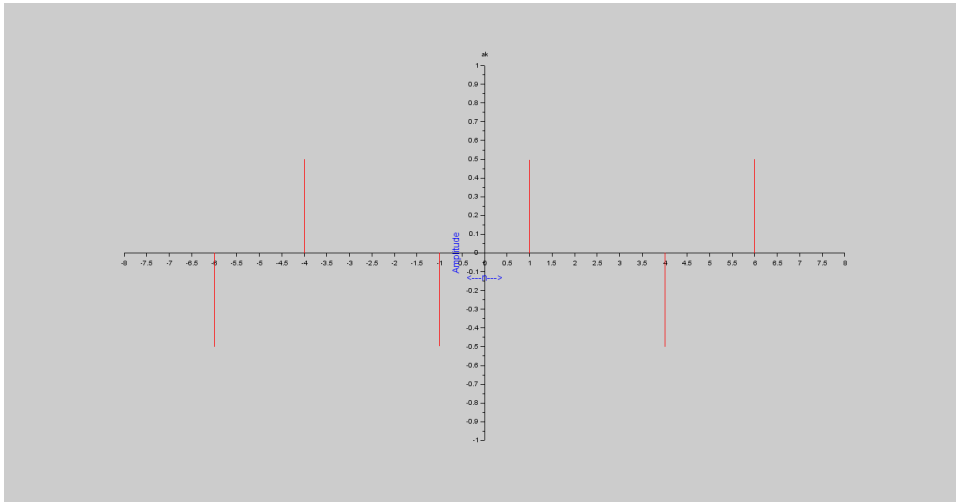


Figure 7.5: Fourier Series of DT Signal

```

27 a.data_bounds = [-8, -1; 8, 1];
28 poly1 = a;
29 poly1.thickness = 3;
30 plot2d3('gnn', k, -imag(ak), 5)
31 xlabel("<---n--->", "fontsize", 3, "color", "blue");
32 ylabel("Amplitude", "fontsize", 3, "color", "blue");
33 poly1 = a;
34 poly1.thickness = 3;
35 plot2d3('gnn', N+k, -imag(ak), 5)
36 poly1 = a.children(1).children(1);
37 poly1.thickness = 3;
38 plot2d3('gnn', -(N+k), -imag(ak($:-1:1)), 5)
39 poly1 = a;
40 poly1.thickness = 3;
41 title('ak')

```

---



# Experiment: 8

## Fourier transform

Scilab code Solution 8.1 Fourier Transform of CT Signal

```
1 // Scilab Version: 6.0.1
2 //O.S : Windows 10
3 //Continuous Time Fourier Transform of  $h(t)=\exp(-At)$ 
    $u(t), t>0$ 
4 clear ;
5 clc ;
6 close ;
7 // Analog Signal
8 A =1; // Amplitude
9 Dt= 0.005;
10 t=0:Dt:10;
11 ht=exp(-A*t);
12 // Continuous time Fourier Transform
13 Wmax=2*%pi*1; // Analog Frequency = 1Hz
14 K=4;
15 k=0:(K/1000):K;
16 W=k*Wmax/K;
17 HW=ht*exp(-sqrt(-1)*t'*W)*Dt;
18 HW_Mag=abs(HW);
19 W=[-mtlb_fliplr(W),W(2:1001)]; // Omega from Wmax to
   Wmax
```

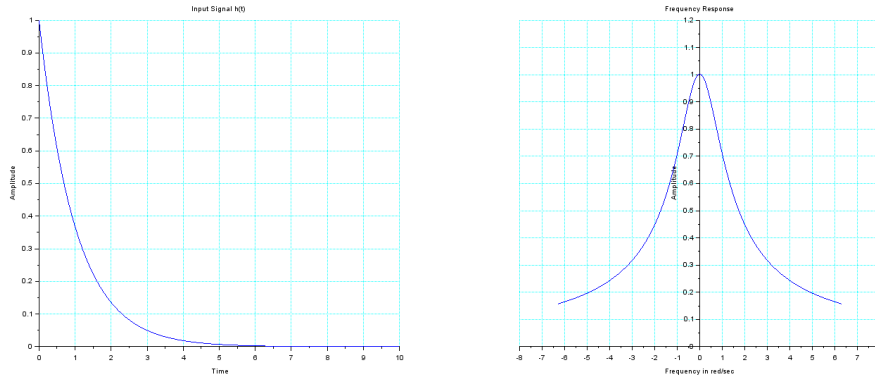


Figure 8.1: Fourier Transform of CT Signal

```

20 HW_Mag=[ mtlbfliplr(HW_Mag),HW_Mag(2:1001)];
21 //Plotting Continuous Time Signal
22 subplot(1,2,1);
23 a=gca();
24 a.y_location="origin";
25 plot(t,ht);
26 xgrid(4,1,7);
27 xlabel("Time");
28 ylabel("Amplitude");
29 title("Input Signal h(t)")
30 //Plotting Magnitude Response of CTS
31 subplot(1,2,2);
32 a=gca();
33 a.y_location="origin";
34 plot(W,HW_Mag);
35 xgrid(4,1,7);
36 xlabel("Frequency in rad/sec");
37 ylabel("Amplitude");
38 title("Frequency Response");

```

---

## Scilab code Solution 8.2 Discrete Time Fourier Transform

```
1 //O.S : Windows 10
2 //Scilab Version: 6.0.1
3 //Caption: Discrete Time Fourier Transform of  $x[n]=$ 
   1 ,  $\text{abs}(n)\leq N1$ 
4 clear;
5 close;
6 clc;
7 // DTS Signal
8 N1 = 2;
9 n=-N1:N1;
10 x=ones(1,length(n));
11 // Discrete-time Fourier Transform
12 Wmax=2*%pi;
13 K =4;
14 k =0:(K/1000):K;
15 W =k*Wmax/K;
16 XW =x*exp(-sqrt(-1)*n'*W);
17 XW_Mag=real(XW);
18 W=[-mtlbfliplr(W), W(2:1001)]; // Omega from -Wmax
   to Wmax
19 XW_Mag=[mtlbfliplr(XW_Mag),XW_Mag(2:1001)];
20 //plot for  $\text{abs}(a)<1$ 
21 subplot(2,1,1);
22 xgrid(4,1,7);
23 xlabel("<-----n----->","fontsize",2,"color", "blue")
   ;
24 ylabel("Amplitude","fontsize",2,"color", "blue");
25 a = gca();
26 a.y_location = "origin";
27 a.x_location = "origin";
28 plot2d3('gmn',n,x);
29 title("Discrete Time Sequence  $x[n]$ ","fontsize",2,"
```

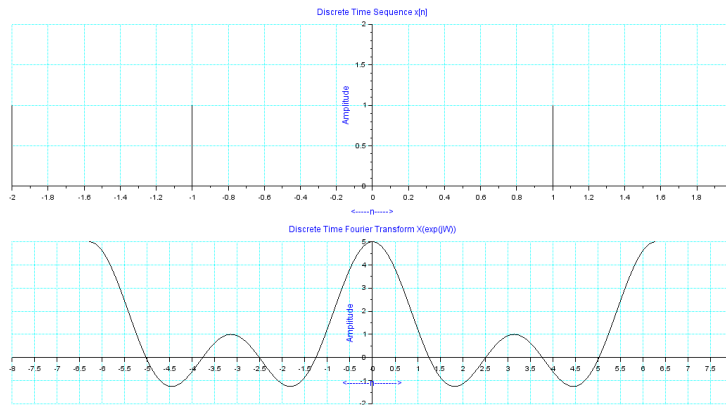


Figure 8.2: Discrete Time Fourier Transform

```

    color", "blue");
30 subplot(2,1,2);
31 xgrid(4,1,7);
32 xlabel("<-----n----->", "fontsize", 2, "color", "
    blue");
33 ylabel("Amplitude", "fontsize", 2, "color", "blue");
34 //ylabel("Amplitude");
35 a = gca();
36 a.y_location = "origin";
37 a.x_location = "origin";
38 plot2d(W, XW_Mag);
39 title("Discrete Time Fourier Transform X(exp(jW))", "
    fontsize", 2, "color", "blue")'

```

---

# Experiment: 9

## Laplace Transform

Scilab code Solution 9.1 Laplace transform

```
1 // Scilab Version: 6.0.1
2 //O.S.: Windows 10
3 //Caption: Laplace transform of  $x(t)=\exp(-0.4t)\cdot\cos(12t)$ 
4 clear;
5 clear all;
6 clc;
7 t=0:0.5:20;
8 ft=exp(-0.4*t).*sin(12*t);
9 s=%s;
10 numfs=s+0.4;
11 denfs=(s+0.4)^2+12^2;
12 fs=syslin('c',numfs/denfs);
13 fs1=csim('impulse',t,fs);
14 disp(fs,"Laplace Transform:–");
15 //disp(fs1,"Laplace Transform:–");
16
17 //Output:
18 //Laplace Transfprm:–
19
20
```

$$\begin{array}{r}
21 // \\
22 // \\
23 // \\
24 //
\end{array}
\frac{0.4 + s}{144.16 + 0.8s + s^2}$$


---

# Experiment: 10

## Z transform & Inverse Z transform

Scilab code Solution 10.1 Z transform and Inverse Z transform of given sequence

```
1 //Expt2: To Z- transform of [1 0 3 -1 2]
2 //O.S: Windows 10;
3 //Scilab: 6.0.1
4 clear;
5 clc ;
6 clear;
7 close ;
8 function [za]=ztransfer(sequence ,n)
9 z=poly(0, 'z', 'r')
10 za=sequence*(1/z)^n'
11 endfunction
12 x1=[1 0 3 -1 2];
13 n=0:length(x1)-1;
14 zz=ztransfer(x1,n);
15 //Display the result in command window
16 disp (zz,"Z-transform of sequence is:");
17 // Expected Output:
18 //Z-transform of sequence is:
```

```

19 //
20 //      2      4
21 //      2 - z + 3z + z
22 //      -----
23 //              4
24 //              z
24 disp('ROC is the entire plane except z = 0');
25 //ROC is the entire plane except z = 0
26
27 //Inverse Z transform
28 z=%z;
29 a =(2+2* z+z ^2) ;
30 b=z^2;
31 h=ldiv(b,a,6);
32 disp(h," The Inverse Z Transform i s ");
33
34 // The Inverse Z Transform is
35 // 1 .
36 // 2 .
37 // 2 .
38 // 0 .
39 // 4 .
40 // 8 .

```

---



# Experiment: 11

## Response of LTI System

This code can be downloaded from the website [www.scilab.in](http://www.scilab.in)

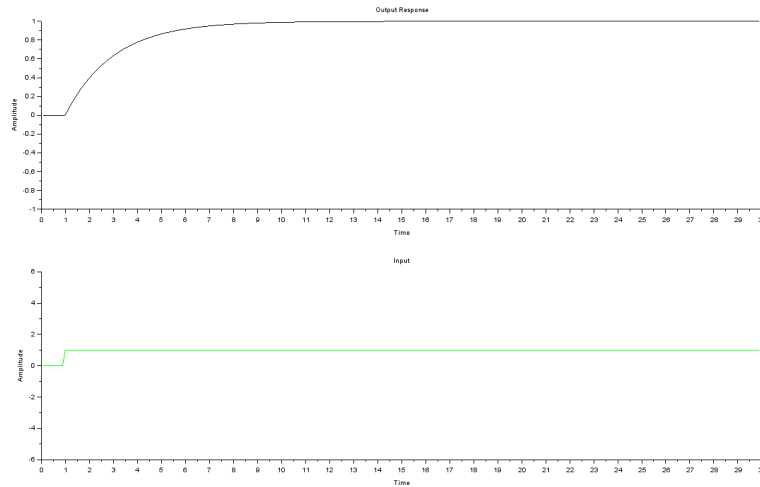


Figure 11.1: Step Response of LTI System

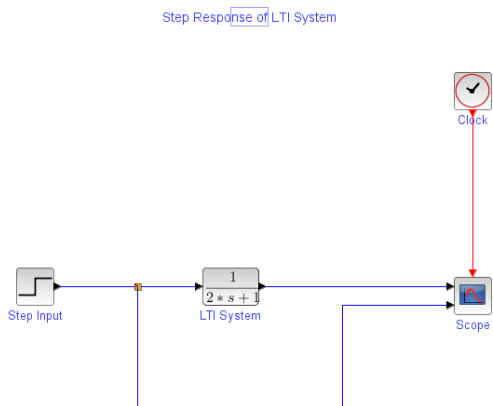


Figure 11.2: Step Response of LTI System

# Experiment: 12

## Probability

Scilab code Solution 12.1 Estimation of probability for tossing of coin

```
1 //Experiment no: 12
2 //Scilab Version: 6.0.1
3 //O.S.: Windows 10
4 //Program to estimate probability of tossing of a
  coin
5 clc;
6 close;
7 clear all;
8 N=input("Number of times coin is to be tossed:")
9 // Input: Number of times coin is to be tossed:1
10 NH=0; // Counter for head
11 NT=0; //counter for tail
12
13 for i=1:N
14     p=grand(1,"prm", (1:2)')';
15
16     if p(1) == 1;
17         NT=NT+1;
18     else
19         NH=NH+1;
20     end
```

```
21 end
22 PH=NH/N;
23 PT=NT/N
24 disp(PH, "Probability of outcome head :");
25 disp(PT,"Probability of outcome tail :");
26
27 // Output:
28 //Probability of outcome head :
29
30 // 1.
31
32 // Probability of outcome tail :
33
34 // 0.
```

---

# Experiment: 13

## Auto correlation

Scilab code Solution 13.1 Auto Correlation of DT signal

```
1 //Experiment no 13 Auto Correlation
2 // SciLab version : 6.0.1
3 // O.S. : Windows 10
4 clear;
5 clc;
6 close;
7 x = input('Enter the given discrete time sequence');
   // Enter a sequence x(n)={1,2,3,4}
8 l = length(x);
9 t=0:l-1;
10 subplot(1,2,1);
11 plot2d3 (t,x,style=[color("blue")]);
12 xgrid(4,1,7);
13 xlabel('n');
14 ylabel('x(n)');
15 title('Input sequence x(n)');
16 h = zeros(1,l);
17 for i = 1:l
18     h(1-i+1) = x(i);
19 end
20 N = 2*l-1;
```

```

21 Rxx = zeros(1,N);
22 for i = 1+1:N
23     h(i) = 0;
24 end
25 for i = 1+1:N
26     x(i) = 0;
27 end
28 for n = 1:N
29     for k = 1:N
30         if(n >= k)
31             Rxx(n) = Rxx(n)+x(n-k+1)*h(k);
32         end
33     end
34 end
35 disp(Rxx,'Auto Correlation Result is ');
36 //Expected Result Rxx(n) = {4,11,20,30,20,11,4}
37 L=length(Rxx);
38 t=0:L-1;
39 subplot(1,2,2);
40 plot2d3 (t,Rxx,style=[color("blue")]);
41 xgrid(4,1,7);
42 xlabel('n');
43 ylabel('Rxx(n)');
44 title('Auto correlation of x(n)');

```

---

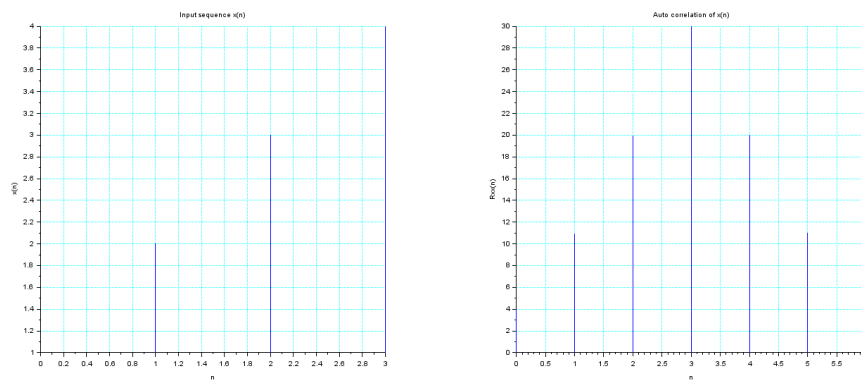


Figure 13.1: Auto Correlation of DT signal

# Experiment: 14

## Cross correlation

Scilab code Solution 14.1 Cross correlation

```
1 //Experiment no 14
2 //cross correlation
3 // SciLab version : 6.0.1
4 // O.S. : Windows 10
5 clc;
6 close ;
7 t1=0:4;
8 x1=[0,1,2,3,4];
9 subplot(2,2,1);
10 xgrid(4,1,7);
11 plot2d3 (t1,x1);
12 xlabel('n');
13 ylabel('x1(n)');
14 title('Input sequence x1(n)');
15
16 t2=0:4;
17 x2=[0,1,5,6,4];
18 subplot(2,2,2);
19 xgrid(4,1,7);
20 plot2d3 (t2,x2);
21 xlabel('n');
```



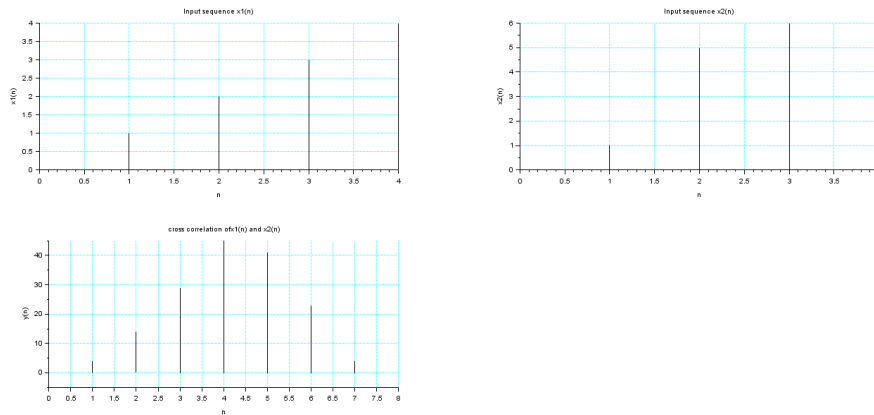


Figure 14.1: Cross correlation

```

22 ylabel('x2(n)');
23 title('Input sequence x2(n)');
24
25 y=xcorr(x1,x2);
26 l=length(y);
27 t3=0:l-1;
28 subplot(2,2,3);
29 plot2d3(t3,y);
30 xgrid(4,1,7);
31 xlabel('n');
32 ylabel('y(n)');
33 title('cross correlation of x1(n) and x2(n)');

```

---