Scilab Manual for Signals and Systems by Prof Manisha Joshi Electronics Engineering VESIT[1]

Solutions provided by
Nandan Hegde
Electronics Engineering
V.E.S.I.T/Mumbai

November 23, 2026

[1]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 To express sum of two complex exponentials as a single sinusoid 5

2 Perform linear convolution sum 9

3 Convolution integral of finite duration signals 12

4 Convolution sum of finite duration sequences 15

5 Frequency Response of Ideal Low pass Filter X (jW) = 1 18

List of Experiments

Solution 1.1	To express sum of two complex exponentials as a single sinusoid	5
Solution 2.1	Perform linear convolution sum	9
Solution 3.1	convolution integral of finite duration signals . .	12
Solution 4.1	convolution sum of finite duration sequences . . .	15
Solution 5.1	Frequency Response of Ideal Low pass Filter . . .	18

List of Figures

1.1 To express sum of two complex exponentials as a single sinusoid 6
1.2 To express sum of two complex exponentials as a single sinusoid 7

2.1 Perform linear convolution sum 10
2.2 Perform linear convolution sum 11

3.1 convolution integral of finite duration signals 13
3.2 convolution integral of finite duration signals 14

4.1 convolution sum of finite duration sequences 16
4.2 convolution sum of finite duration sequences 17

5.1 Frequency Response of Ideal Low pass Filter 19
5.2 Frequency Response of Ideal Low pass Filter 20

Experiment: 1

To express sum of two complex exponentials as a single sinusoid

Scilab code Solution 1.1 To express sum of two complex exponentials as a single sinusoid

```
1 //To express sum of two complex exponentials as a
     single sinusoid
2 //scilab 5.4.1 ;64 bit(windows 8)
3 clear;
4 clc;
5 close;
6 t =0:1/100:2*%pi;
7 x1 = exp(sqrt(-1)*2*t);
8 x2 = exp(sqrt(-1)*3*t);
9 x = x1+x2;
10 for i = 1:length(x)
11   X(i) = sqrt((real(x(i)).^2)+(imag(x(i)).^2));
12 end
13 plot(t,X);
```

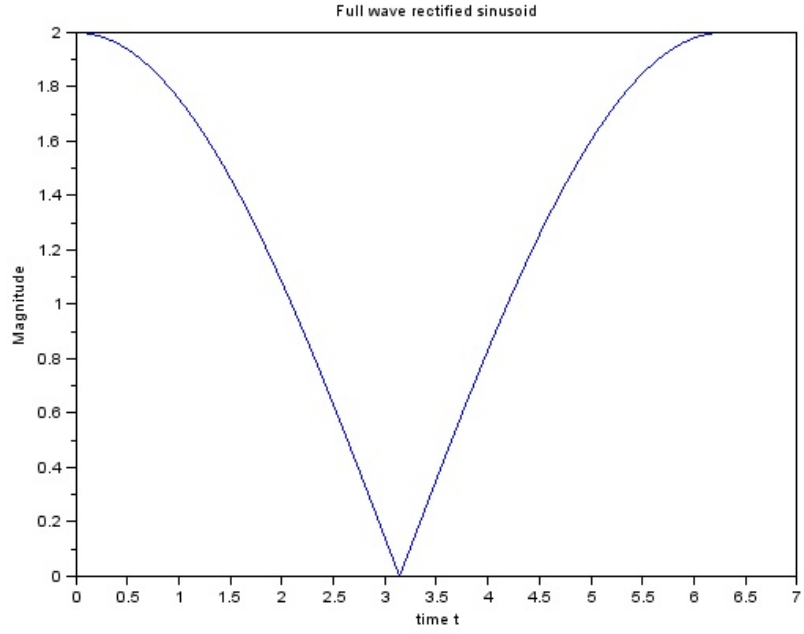

Figure 1.1: To express sum of two complex exponentials as a single sinusoid

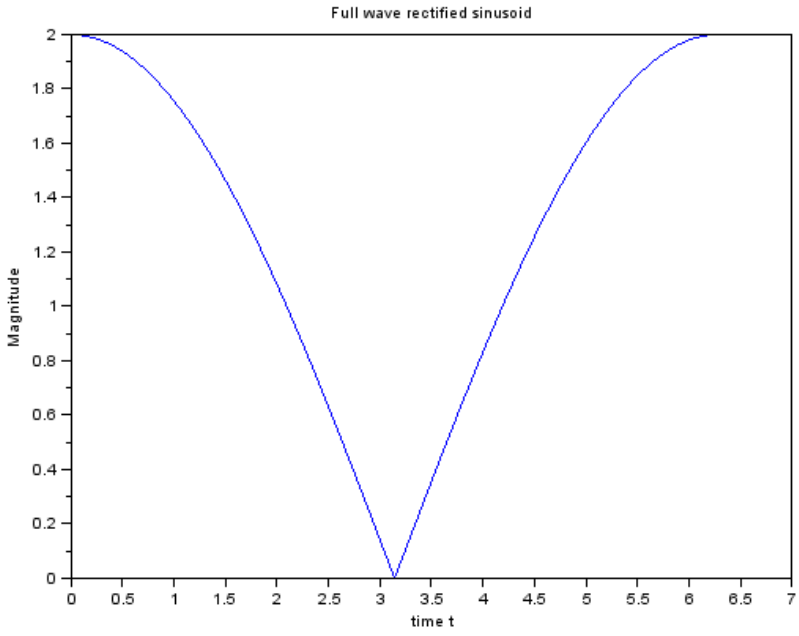

Figure 1.2: To express sum of two complex exponentials as a single sinusoid

```
14 xtitle('Full wave rectified sinusoid','time t','
     Magnitude');
```

Experiment: 2

Perform linear convolution sum

Scilab code Solution 2.1 Perform linear convolution sum

```
1  // To Perform linear convolution sum
2  //scilab 5.4.1 ;64 bit(windows 8)
3  clear;
4  close;
5  clc;
6  h = [0,0,1,1,1,0,0];
7  N1 = -2:4;
8  x = [0,0,0.5,2,0,0,0];
9  N2 = -2:4;
10 y = convol(x,h);
11 for i = 1:length(y)
12   if (y(i)<=0.0001)
13     y(i)=0;
14   end
15 end
16 N = -4:8;
17 subplot(3,1,1)
18 a=gca();
19 plot2d3('gnn',N1,h)
20 xtitle('Impulse Response','n','h[n]');
21 a.thickness = 2;
```

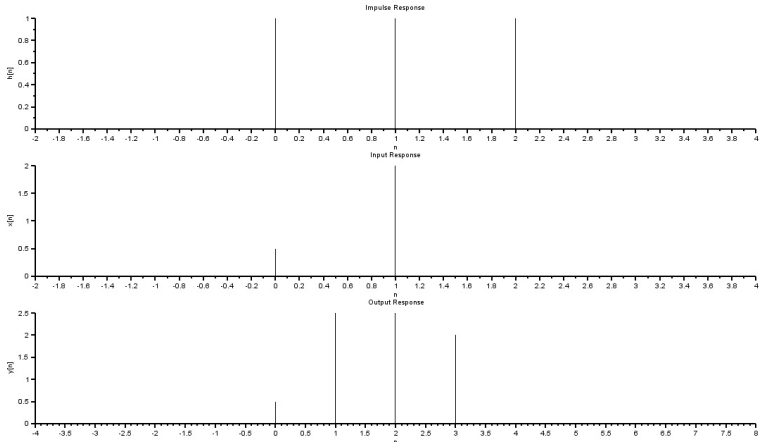

Figure 2.1: Perform linear convolution sum

```
22 subplot(3,1,2)
23 a=gca();
24 plot2d3('gnn',N2,x)
25 xtitle('Input Response','n','x[n]');
26 a.thickness = 2;
27 subplot(3,1,3)
28 a=gca();
29 plot2d3('gnn',N,y)
30 xtitle('Output Response','n','y[n]');
31 a.thickness = 2;
```

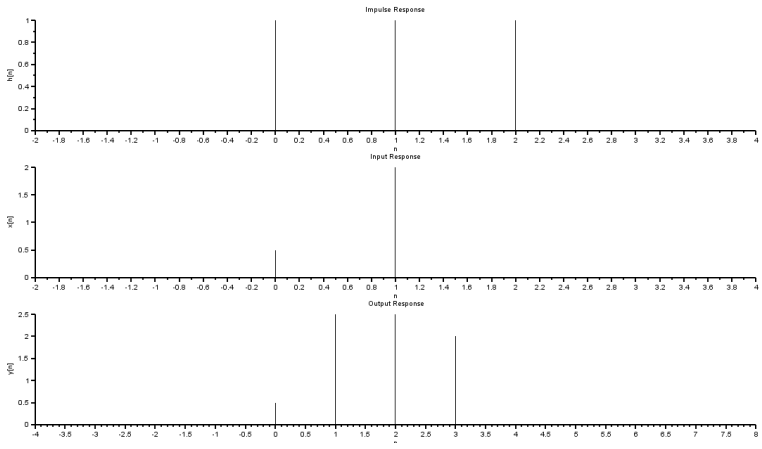

Figure 2.2: Perform linear convolution sum

Experiment: 3

Convolution integral of finite duration signals

Scilab code Solution 3.1 convolution integral of finite duration signals

```
1  //Convolution Integral of fintie duration signals
2  //scilab 5.4.1 ;64 bit(windows 8)
3  clear;
4  close;
5  clc;
6  T = 10;
7  x = ones(1,T);  //Input Response
8  for t = 1:2*T
9    h(t) = t-1;    //Impulse Response
10 end
11 N1 = 0:length(x)-1;
12 N2 = 0:length(h)-1;
13 y = convol(x,h);
14 N = 0:length(x)+length(h)-2;
15 subplot(3,1,1)
16 a=gca();
17 a.x_location="origin";
18 plot2d(N2,h)
19 xtitle('Impulse Response','t','h(t)');
```

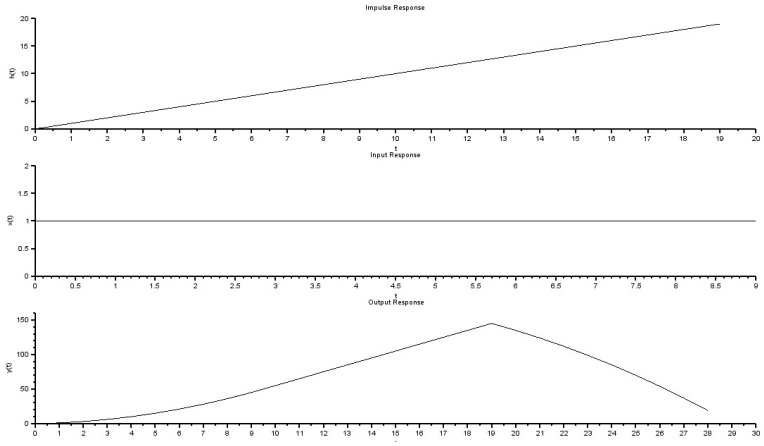

Figure 3.1: convolution integral of finite duration signals

```
20 a.thickness = 2;
21 subplot(3,1,2)
22 a=gca();
23 plot2d(N1,x)
24 xtitle('Input Response','t','x(t)');
25 a.thickness = 2;
26 subplot(3,1,3)
27 a=gca();
28 plot2d(N,y)
29 xtitle('Output Response','t','y(t)');
30 a.thickness = 2;
```

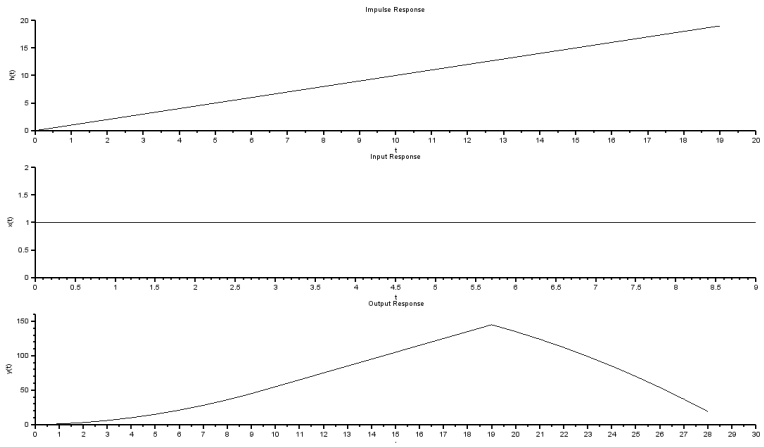

Figure 3.2: convolution integral of finite duration signals

Experiment: 4

Convolution sum of finite duration sequences

Scilab code Solution 4.1 convolution sum of finite duration sequences

```
1  //Convolution Sum of finite duration sequences
2  //scilab 5.4.1 ;64 bit(windows 8)
3  clear;
4  close;
5  clc;
6  x = ones(1,5);
7  N1 =0:length(x)-1;
8  Alpha = 1.4;     //alpha > 1
9  for n = 1:7
10   h(n)= (Alpha^(n-1))*1;
11 end
12 N2 =0:length(h)-1;
13 y = convol(x,h);
14 N = 0:length(x)+length(h)-2;
15 subplot(3,1,1)
16 a=gca();
17 plot2d3('gnn',N2,h)
18 xtitle('Impulse Response','n','h[n]');
19 a.thickness = 2;
```

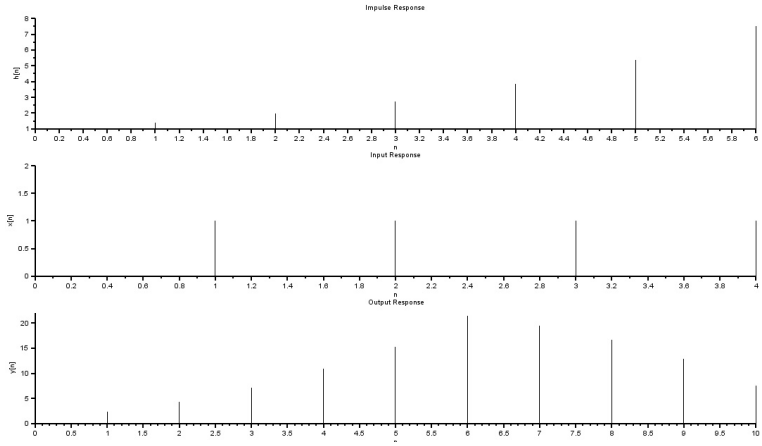

Figure 4.1: convolution sum of finite duration sequences

```
20 subplot(3,1,2)
21 a=gca();
22 plot2d3('gnn',N1,x)
23 xtitle('Input Response','n','x[n]');
24 a.thickness = 2;
25 subplot(3,1,3)
26 a=gca();
27 plot2d3('gnn',N,y)
28 xtitle('Output Response','n','y[n]');
29 a.thickness = 2;
```

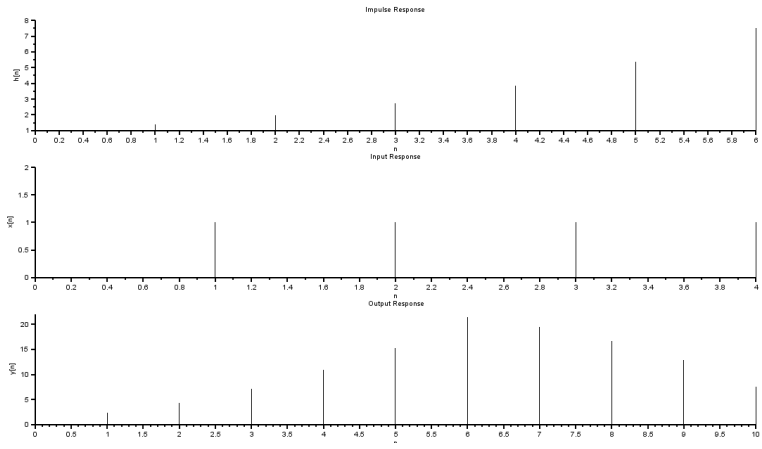

Figure 4.2: convolution sum of finite duration sequences

Experiment: 5

Frequency Response of Ideal Low pass Filter X (jW) = 1

Scilab code Solution 5.1 Frequency Response of Ideal Low pass Filter

```
1 //Frequency Response of Ideal Low pass Filter ,X(jW)=
      1
2 //scilab 5.4.1 ;64 bit(windows 8)
3 clear;
4 clc;
5 close;
6 Wc = 10;     //1 rad/sec
7 W  = -Wc:0.1:Wc; //Passband of filter
8 HW0 = 1; //Magnitude of Filter
9 HW = HW0*ones(1,length(W));
10 //Inverse Continuous-time Fourier Transform
11 t = -%pi:%pi/length(W):%pi;
12 Dw = 0.1;
13 ht =(1/(2*%pi))*HW *exp(sqrt(-1)*W'*t)*Dw;
14 ht = real(ht);
```

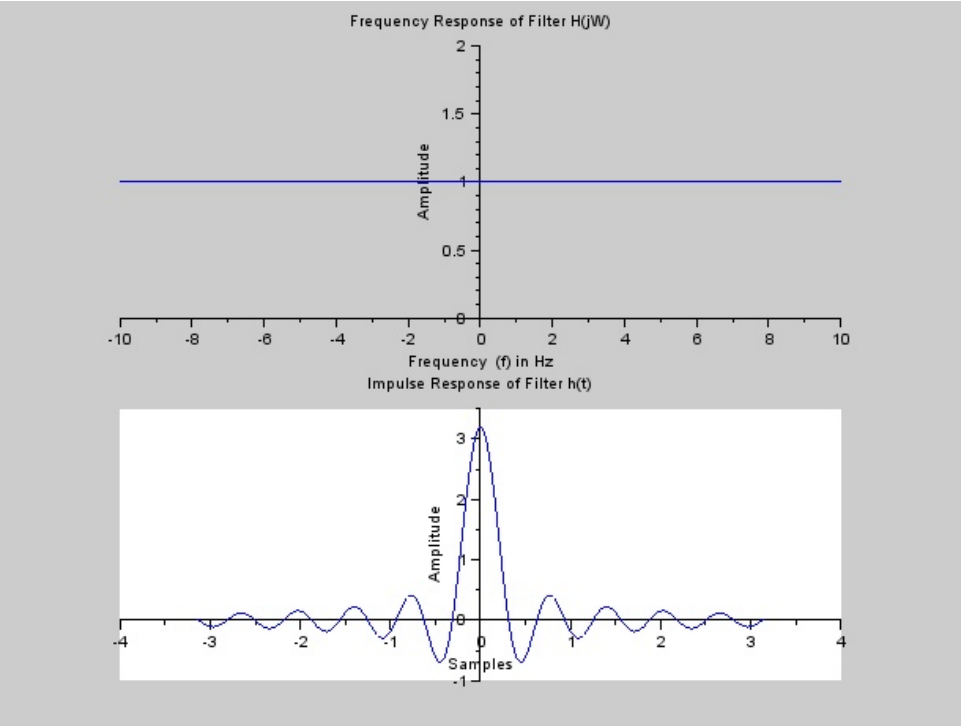

Figure 5.1: Frequency Response of Ideal Low pass Filter

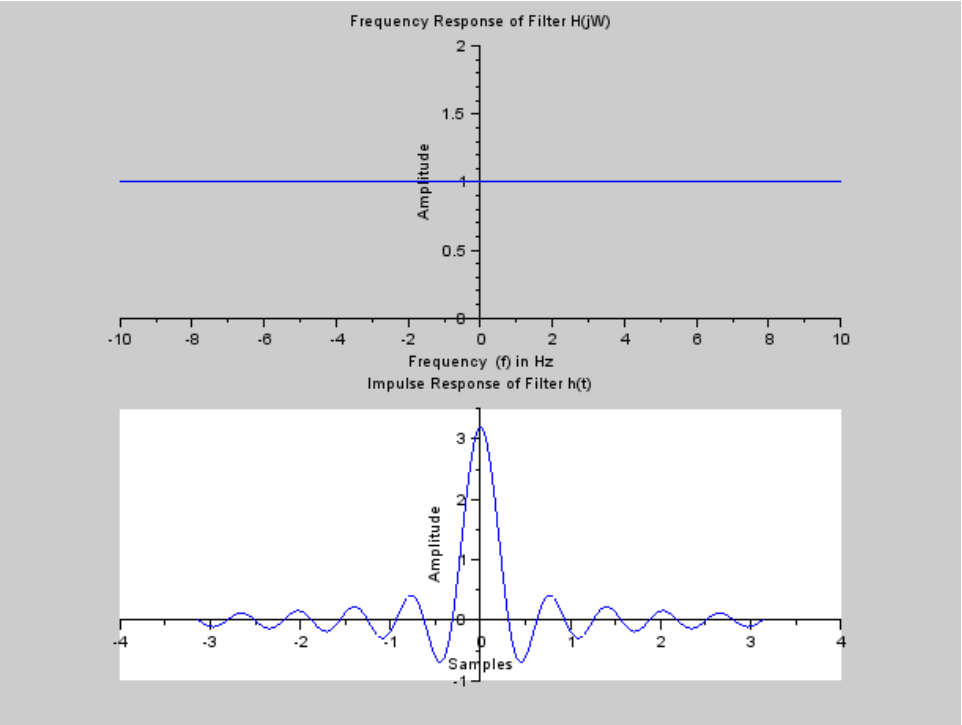

Figure 5.2: Frequency Response of Ideal Low pass Filter

```
15 subplot(2,1,1)
16 a = gca();
17 a.y_location ="origin";
18 a.x_location ="origin";
19 plot(W,HW);
20 xtitle('Frequency Response of Filter H(jW)')
21 xlabel('Frequency (f) in Hz');
22 ylabel('Amplitude');
23 subplot(2,1,2)
24 a = gca();
25 a.y_location ="origin";
26 a.x_location ="origin";
27 plot(t,ht);
28 xtitle('Impulse Response of Filter h(t)')
29 xlabel('Samples');
30 ylabel('Amplitude');
```