Scilab Manual for Digital Signal Processing Lab

by Dr Veena Hegde
Instrumentation Engineering
B.M.S College Of Engineering[1]

Solutions provided by
Dr Veena Hegde
Instrumentation Engineering
B.m.s College Of Engineering

July 16, 2024

[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 Experiments

List of Figures

Experiment: 1

Design and Testing of a Digital Butterworth Low pass filter with cutoff of 5 KHz to filter an input (.wav) file

Scilab code Solution 1.1 DSP Lab Migration

```

// Exp 1: Design Low Pass Filter as per the given specification and test the working by taking an input sound signal.
// Enter cutoff freq in Hz fc = 5

// Version : Scilab 5.2.2
// Operating Syatem : Ubuntu 16.04 LTS

clc;                    //clear console
clear;
xdel(winsid());
fc=input('Enter cutoff freq in Hz fc =')        //Cutoff frequency
```

```
fs=11025;
n=11;  //Filter order
Fp=2*fc/fs;
[Hz]=iir(n,'lp','butt',[Fp/2,0],[0,0])
[p,z,g]=iir(n,'lp','butt',[Fp/2,0],[0,0])  //Filter design
[Hw,w]=frmag(Hz,256);
figure(1)
subplot(2,1,1)
plot(2*w,abs(Hw));
xlabel('Normalized Digital frequency w−>')
ylabel('magnitude');
title('Magnitude response of IIR filter')
xgrid(1)
subplot(2,1,2)
plot(2*w*fs,abs(Hw));
xlabel('Analog Frequency in Hz f −−−>')
ylabel('Magnitude |H(w)|=')
title('Magnitude Response of IIR LPF')
xgrid(1)

[y,Fs]=wavread("meow.wav")  //Reading input sound signal
figure(2)
subplot(2,1,1)
plot(y)
title('Input signal waveform');
xlabel('Frequency−−>');
ylabel('Magnitude−−>');
playsnd(y)

outlo=filter(z,abs(p),y);  //Passing accquired signal through desired filter
subplot(2,1,2)
plot(outlo)
title('Output signal waveform after filtering')
xlabel('Frequency−−>');
```

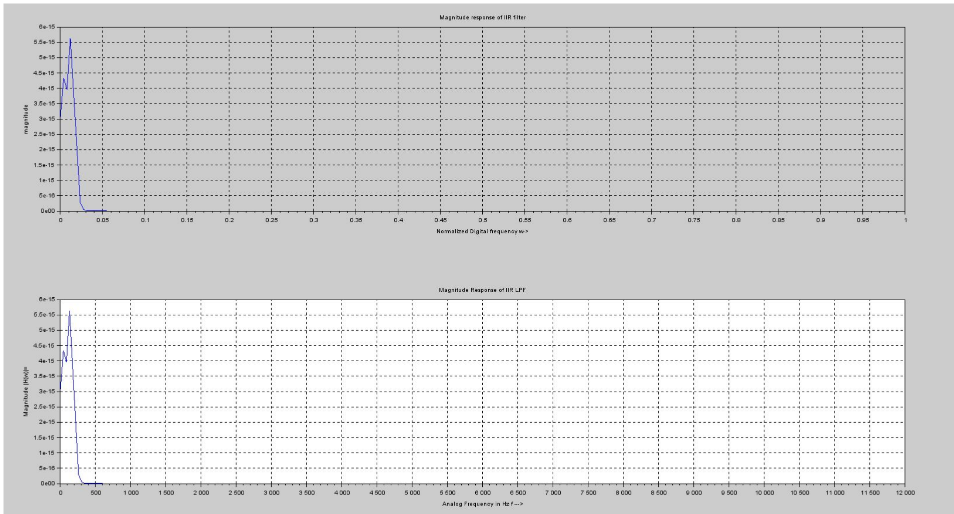

Figure 1.1: DSP Lab Migration

```
ylabel('Magnitude—>');
```

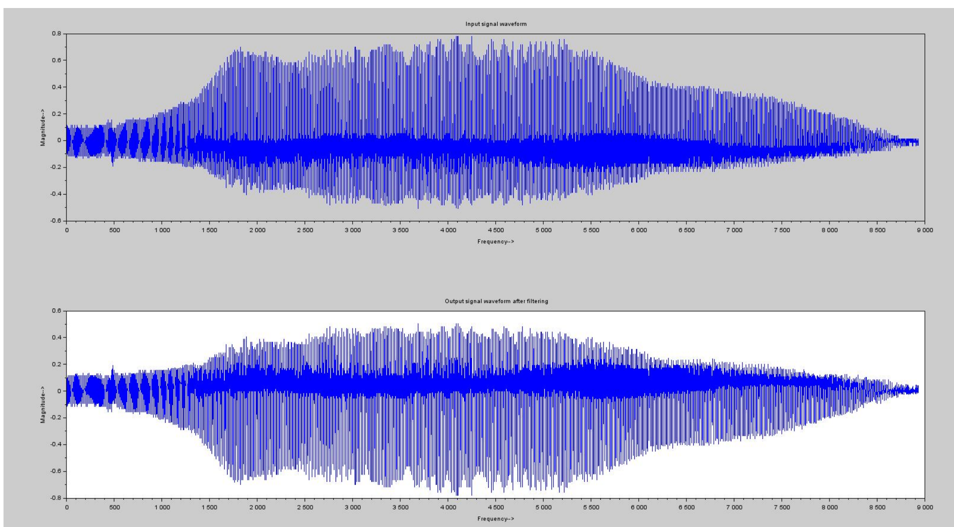

Figure 1.2: DSP Lab Migration

Experiment: 2

Design a digital Butterworth low pass filter to band limit a sine wave up to 4 KHz, by considering input as 'tone'

Scilab code Solution 2.2 DSP Lab Migration

```
//Program to design a Butterworth Low pass filter to
    Band limit a sine wave up to 4kHz . Taking the
   input as a     tone      .
// Version : Scilab 5.2.2
// Operating Syatem : Ubuntu 16.04 LTS
clc;
clear;
xdel(winsid());
Fc =4000;                                          // Cut-
   off frequency
Fs =44100;                                         //
   Sampling frequency
N =8 ;                                             // Order
Fp = 2*Fc/Fs;                                      //Pass
   band edge frequency
```

```
[Hz]=iir(N,'lp','butt',[Fp/2,0],[0,0])
[p,z,g]=iir(N,'lp','butt',[Fp/2,0],[0,0])    //
    digital IIR Butterworth Filter
[Hw,w] = frmag(Hz ,256);

// Plotting the filter design
figure(1)
plot(2*w,abs(Hw));
xlabel("Digital Frequency  Normalized (w)")
ylabel("Magnitude")
title("Magnitude Response of Butterworth filter ")
xgrid(1)

[y,Fs]=wavread("tone1k.wav")                    //
    Reading the y
figure(2)
subplot(2,1,1)
plot(y)
title('Input signal waveform before filtering');
xlabel('Frequency');
ylabel('Magnitude');
playsnd(y)
L=length(y)

outlow=filter(z,abs(p),y);
subplot(2,1,2)
plot(outlow)
title('Output signal waveform after filtering')
xlabel('Frequency');
ylabel('Magnitude');

playsnd(outlow)

Da = fft(y,-1);
Pyy = (1/L)*(abs(Da).^2); //Peridogram Estimate
```

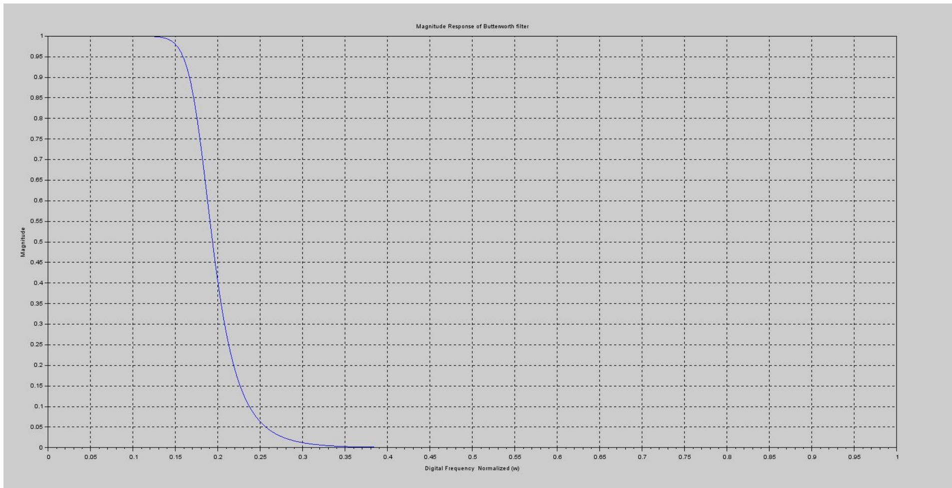

Figure 2.1: DSP Lab Migration

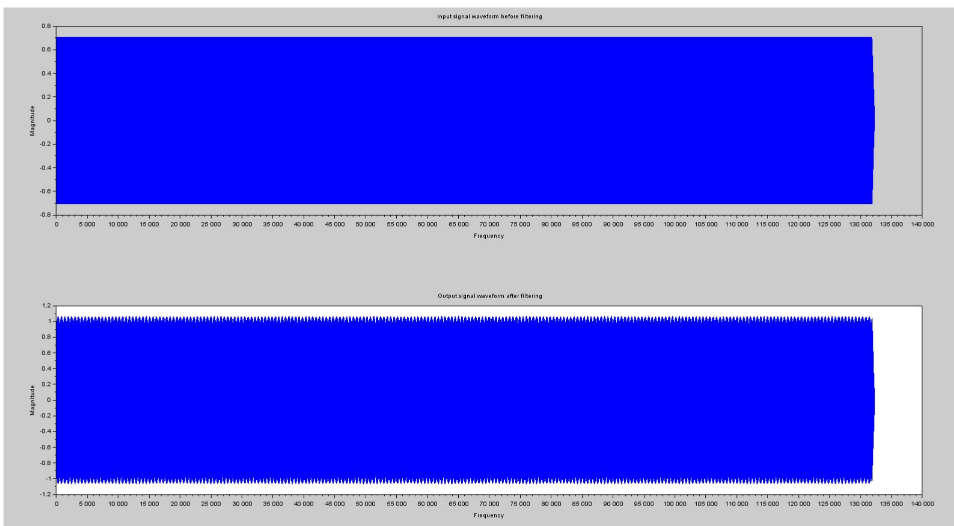

Figure 2.2: DSP Lab Migration

Experiment: 3

Illustrate the working of a digital low pass filter by taking audio data as input and allow frequency up to 2 KHz

Scilab code Solution 3.3 DSP Lab Migration

```
// Design a filter using Butterworth polynomial for
   the following specifications:
//1.    Order : 7
//2.    Cut-off frequency : 2.5 kHz

// Version : Scilab 5.2.2
// Operating Syatem : Ubuntu 16.04 LTS

clc;
clear;
xdel(winsid());
Fc =2500;

   // Cut-off frequency
Fs =44100;
```

```
    // Sampling frequency
N =7 ;
    // Order of the filter
Fp = 2*Fc/Fs;
    //Pass band edge frequency
[Hz]=iir(N,'lp','butt',[Fp/2,0],[0,0])
[p,z,g]=iir(N,'lp','butt',[Fp/2,0],[0,0])
[Hw,w] = frmag(Hz ,256);

figure(1)
plot(2*w,abs(Hw));
xlabel("Digital Frequency  Normalized (w)")
ylabel("Magnitude")
title("Magnitude Response of Butterworth filter ")
xgrid(1)

[y,Fs]=wavread("tone1k.wav")
    // Reading the .wav file

outlow=filter(z,abs(p),y);

psd1=pspect(100,200,'re',y)
figure(2)
subplot(2,1,1)
plot(psd1)
    //Plotting power spectral density of input
title('Input signal power spectral density')
xlabel('Frequency');
ylabel('Magnitude');

psd2=pspect(100,200,'re',outlow)
subplot(2,1,2)
plot(psd2)
```

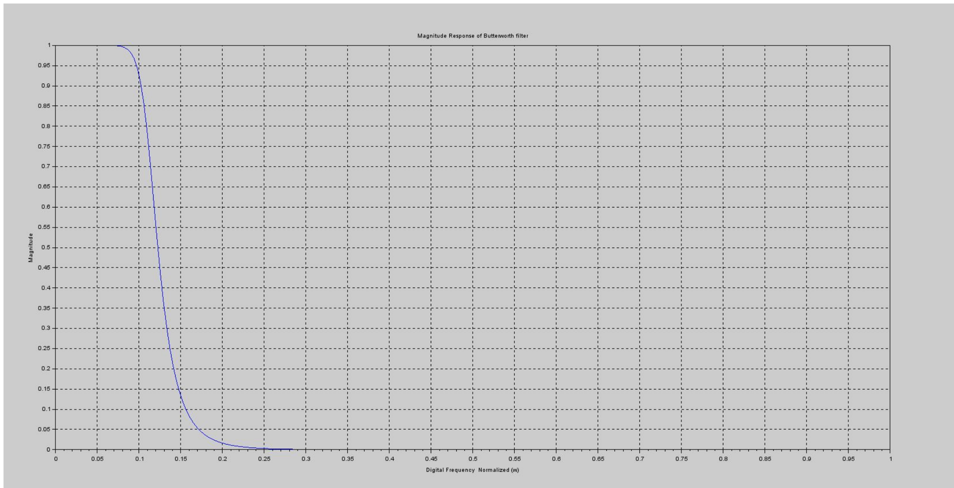

Figure 3.1: DSP Lab Migration

```
//Plotting power spectral density of output
title('Filtered signal power spectral density')
xlabel('Frequency');
ylabel('Magnitude');

playsnd(outlow)
```

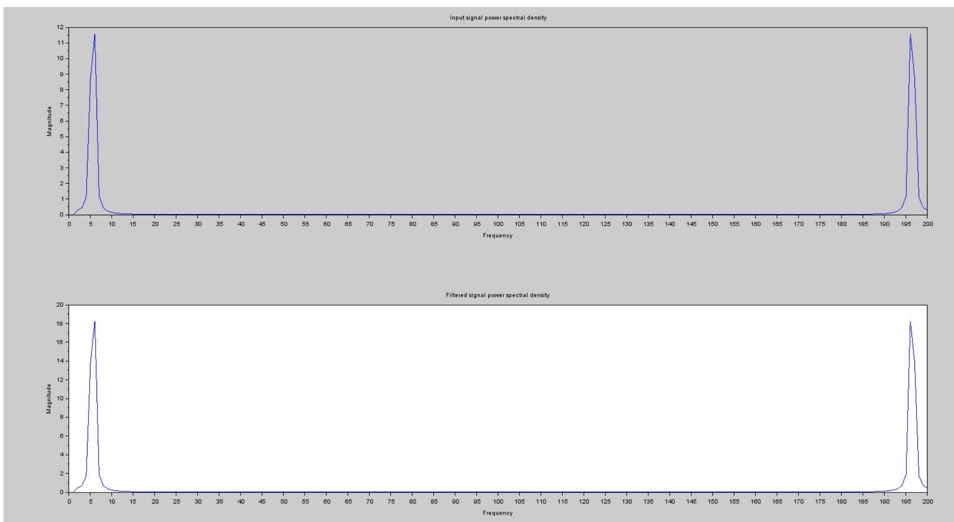

Figure 3.2: DSP Lab Migration

Experiment: 4

Design and test the working of a High Pass FIR filter using Hamming window by taking a high frequency signal as input

Scilab code Solution 4.4 DSP Lab Migration

```
// Version : Scilab 5.2.2
// Operating Syatem : Ubuntu 16.04 LTS

clc;        clear;      xdel(winsid());
fc=20000;   fs=44100;   M=63;                                   //
    Filter order
wc=2*fc/fs;
[wft,wfm,fr]=wfir('hp',M,[wc/2,0],'hm',[0,0]);
    //Fir Filter
figure(1)
```

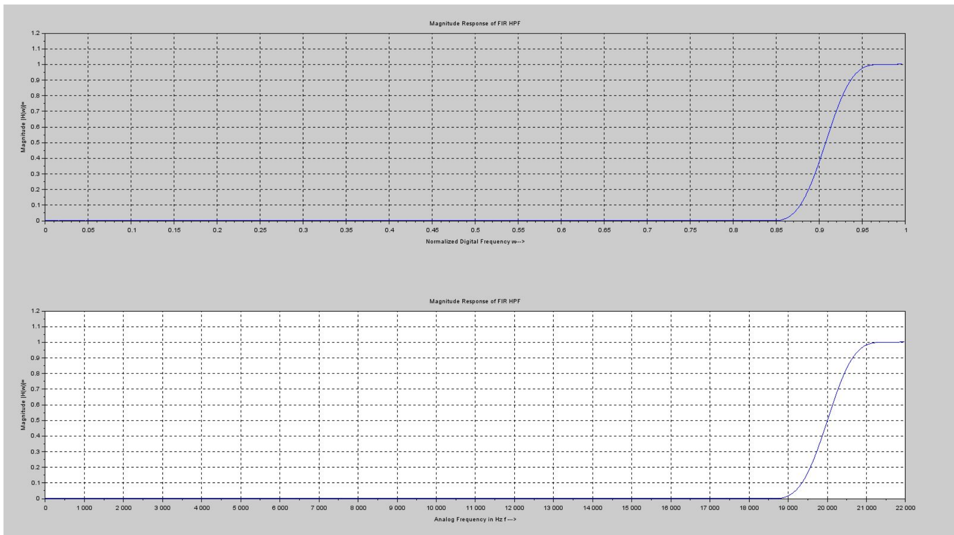

Figure 4.1: DSP Lab Migration

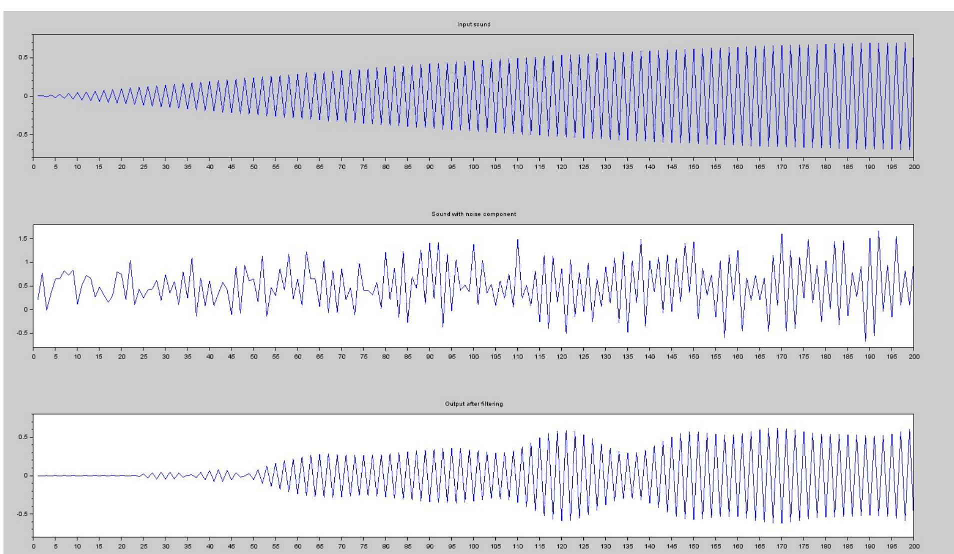

Figure 4.2: DSP Lab Migration

```
subplot(2,1,1);          plot(2*fr,wfm);
xlabel('Normalized Digital Frequency w--->')
ylabel('Magnitude |H(w)|=')
title('Magnitude Response of FIR HPF');      xgrid(1)
subplot(2,1,2);          plot(fr*fs,wfm);
xlabel('Analog Frequency in Hz f --->')
ylabel('Magnitude |H(w)|=')
title('Magnitude Response of FIR HPF');      xgrid(1)

[d,Fs]=wavread("22000.wav")
playsnd(d,Fs)        //single tone high frequency sound wave
L = length(d);       a=1+nextpow2(L);        N=2*(2^a);
noise = rand(1,L);          data = d+noise;     playsnd(data);
outhi = filter(wft,1,data);  playsnd(outhi);

figure(2)
subplot(3,1,1);      plot(d(1:200));
title('Input sound');
subplot(3,1,2);      plot(data(1:200));
title('Sound with noise component');
subplot(3,1,3);      plot(outhi(1:200));
title('Output after filtering');
```

Experiment: 5

Design and test the working of a Butterworth band pass filter by giving a time domain input signal

Scilab code Solution 5.5 DSP lab migration

```
// Version : Scilab 5.2.2
// Operating Syatem : Ubuntu 16.04 LTS
//Assume fp=1000 hz and fs=5000 hz
//Use the formula to convert the pass band ripple
    and To convert stop band attenuation in dB(A in
    dB)=-20*log10(Ap or As)
// assume kp=1.93 dB and ks=13.97 dB


clc;
clear;
xdel(winsid());
fp= input( " Enter the pass band edge (Hz) = ");
```

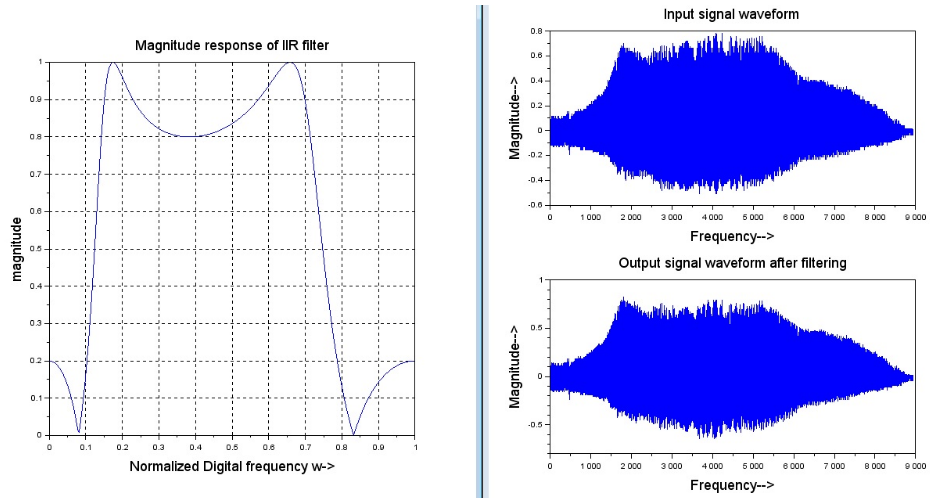

Figure 5.1: DSP lab migration

```
fs= input("Enter the stop band edge (Hz) = ");
kp= 1.93                //assume the pass band ripple(dB)
ks= 13.97               //assume the stop band attenuation(dB)
Fsf=44000;              //sampling frequency
//Converting to digital frequency
Fp1=2*3.14*fp/Fsf;
Fs1=2*3.14*fs/Fsf;

// Digital filter specifications ( rad/samples )
N = log10(sqrt((10^(0.1*ks)-1)/(10^(0.1*kp)-1)))/log10(Fs1/Fp1);  //Order of the filter
N = ceil(N);               //rounded to nearest integer
disp(N," IIR Filter order N =");

oc = 0.5*((Fp1*Fsf)/((10^(0.1*kp)-1)^(1/(2*N))) + (Fs1*Fsf)/((10^(0.1*ks)-1)^(1/(2*N))) );  //Cut off Frequency
disp(oc, "Cutoff Frequency in rad/ seconds OC =")
[Hz]=iir(N,'bp','ellip',[Fp1/2,Fs1/2],[0.2,0.200])
```

```
  //the sum of last matrix [0.2,0.200] must be less than 1
[p,z,g]=iir(N,'bp','ellip',[Fp1/2,Fs1/2],[0.2,0.200])
[Hw,w]=frmag(Hz,256);
figure(1)
plot(2*w,abs(Hw));
xlabel('Normalized Digital frequency w-->')
ylabel('magnitude');
title('Magnitude response of IIR filter')
xgrid(1)
[y,Fs]=wavread("H:\DSP SCILAB\Final\meow.wav")    //Reading input .wav signal file path of the .wav file must be changed
figure(2)
subplot(2,1,1)
plot(y)
title('Input signal waveform');
xlabel('Frequency---->');
ylabel('Magnitude---->');
playsnd(y)
outlo=filter(abs(z),abs(p),y);      //Passing accquired signal through desired filter
subplot(2,1,2)
plot(outlo)
title('Output signal waveform after filtering')
xlabel('Frequency---->');
ylabel('Magnitude---->');

N =length(y);                               //Power spectral density of the Input signal
Y = fft(y,-1);
Pxx = (1/N)*(abs(Y).^2); //Peridogram Estimate
figure(3)
plot2d3('gnn',[1:N],Pxx)
title('Input signal power spectral density')
xlabel('Analog Frequency in Hz f ------>')
ylabel('Magnitude |H(w)|=')
```

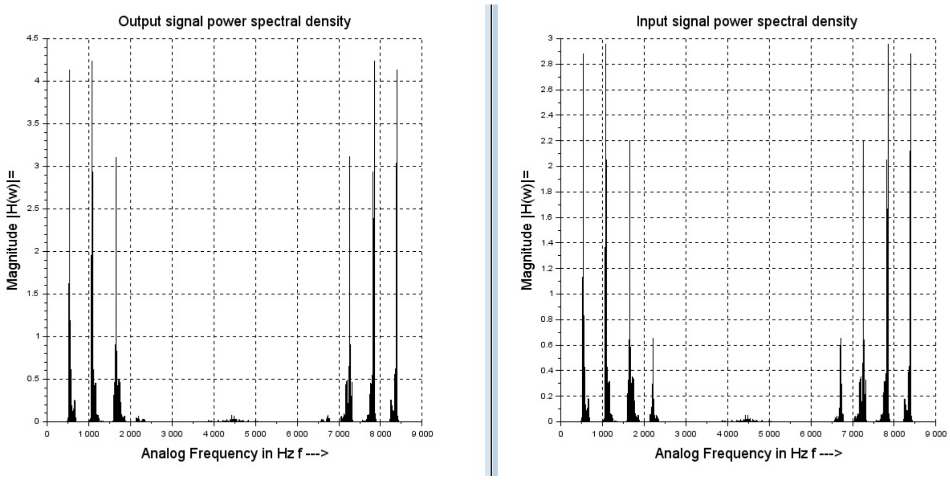

Figure 5.2: DSP lab migration

```
xgrid(1)
playsnd(outlo)

N=length(outlo)                                  ////
   Power spectral density of the Ouput signal
OL = fft(outlo,-1);
Fxx = (1/N)*(abs(OL).^2); //Peridogram Estimate
figure(4)
plot2d3('gnn',Fxx)
title('Output signal power spectral density')
xlabel('Analog Frequency in Hz f --->')
ylabel('Magnitude |H(w)|=')
xgrid(1)
```

Experiment: 6

Design and test band pass, Butterworth filter for given specifications which includes Edge frequencies, Ripple and Attenuation

Scilab code Solution 6.6 DSP lab migration

```
// Version : Scilab 5.2.2
// Operating Syatem : Ubuntu 16.04 LTS

clc;
clear;
xdel(winsid());
fp= input( " Enter the pass band edge (Hz) = ");
fs= input( "Enter the stop band edge (Hz) = ");
kp= input( "Enter the pass band attenuation (dB) =")
```

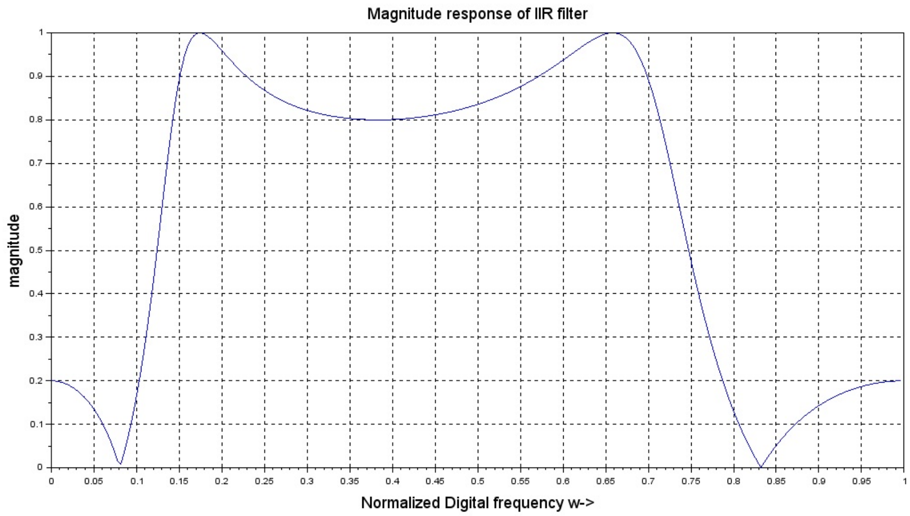

Figure 6.1: DSP lab migration

```
;
ks= input( "Enter the stop band attenuation (dB) = ") ;
//kp= 1.93                //assume the pass band ripple (dB)
//ks= 13.97               //assume the stop band attenuation(dB)
Fsf=44000;            //sampling frequency
//Converting to digital frequency
Fp1=2*3.14*fp/Fsf;
Fs1=2*3.14*fs/Fsf;

// Digital filter specifications ( rad/samples )
N = log10(sqrt((10^(0.1*ks)-1)/(10^(0.1*kp)-1)))/log10(Fs1/Fp1);  //Order of the filter
N = ceil(N);              //rounded to nearest integer
disp(N," IIR Filter order N =");

oc = 0.5*((Fp1*Fsf)/((10^(0.1*kp)-1)^(1/(2*N))) + (Fs1*Fsf)/((10^(0.1*ks)-1)^(1/(2*N))) );  //Cut
```

```
 Enter the pass band edge (Hz) =
1000


Enter the stop band edge (Hz) =
5000

Enter the pass band attenuation (dB) =
2

Enter the stop band attenuation (dB) =
14


  IIR Filter order N =

   2.

 Cutoff Frequency in rad/ seconds OC =
   10675.066
```

Figure 6.2: DSP lab migration

```
off Frequency
disp(oc, "Cutoff Frequency in rad/ seconds OC =")
[Hz]=iir(N,'bp','ellip',[Fp1/2,Fs1/2],[0.2,0.200]) //the sum of last matrix [0.2,0.200] must be less than 1
[p,z,g]=iir(N,'bp','ellip',[Fp1/2,Fs1/2],[0.2,0.200])
[Hw,w]=frmag(Hz,256);
figure(1)
plot(2*w,abs(Hw));
xlabel('Normalized Digital frequency w->')
ylabel('magnitude');
title('Magnitude response of IIR filter')
xgrid(1)
```