EXPERIMENT – 5
DSP – LAB

NAME – AMAN KUMAR
ROLL NO – EE22BTECH11006

AIM – To design digital filter such as Low Pass Filter (LPF) and Band Pass Filter (BPF).

Theory:

FIR filter means Finite Impulse Response and its impulse response is of finite period, as a result it settles down to zero in finite time and it doesn't have a feedback look like IIR filter as it has infinite impulse response.

In this type of filter, the digital signals by convolving the input with a finite set of coefficients, producing an output. The FIR filters offer stability, linear phase response, and flexible frequency shaping making them suitable for diverse signal processing. One of the design methods includes windowing in which we have a windowing function such as hamming window function and we multiply the desired impulse response and we get our h[n] and thus this method helps in creation of filter with finite impulse response and customizable frequency response.

<u>LPF (Low Pass Filter):</u>

$$h_d[n] = \begin{cases} \sin(\omega_c n)/(\pi n), & -(N-1)/2 \le n \le (N-1)/2 \\ \omega_c/\pi, & n = 0 \end{cases}$$

$$w_h[n] = \begin{cases} 0.54 - 0.46 \cos(2\pi n/(N-1)), & \text{if } 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

MATLAB code:

```
fc = 400;
wc=pi/2;
N=39;
fs=2*pi*fc/wc;
h_n= lowpassfilter(fc,fs,N);
disp(h_n);
stem(h_n);
fvtool(h_n);
function h_n = lowpassfilter(fc,fs,N)
    wc=2*pi*fc/fs;
    h_d = zeros(1,N);
    for n = -(N-1)/2:(N-1)/2
        if n == 0
            h_d(n+(N-1)/2 +1) = wc/pi;
        else
            h_d(n+(N-1)/2+1) = sin(wc*n)/(pi*n);
        end
    end
    w_h = zeros(1,N);
    for k= 0:N-1
        w_h(k+1) = 0.54-0.46*cos(2*pi*k/(N-1));
    end
    h_n = h_d.*w_h;
end
```

RESULT:

```
>> lpf

 Columns 1 through 10

  -0.0013   0.0000   0.0020  -0.0000  -0.0038   0.0000   0.0071  -0.0000  -0.0124
0.0000

 Columns 11 through 20

   0.0204  -0.0000  -0.0330   0.0000   0.0542  -0.0000  -0.1002   0.0000   0.3163
0.5000

 Columns 21 through 30

   0.3163   0.0000  -0.1002  -0.0000   0.0542   0.0000  -0.0330  -0.0000   0.0204
0.0000

 Columns 31 through 39

  -0.0124  -0.0000   0.0071   0.0000  -0.0038  -0.0000   0.0020   0.0000  -0.0013
```

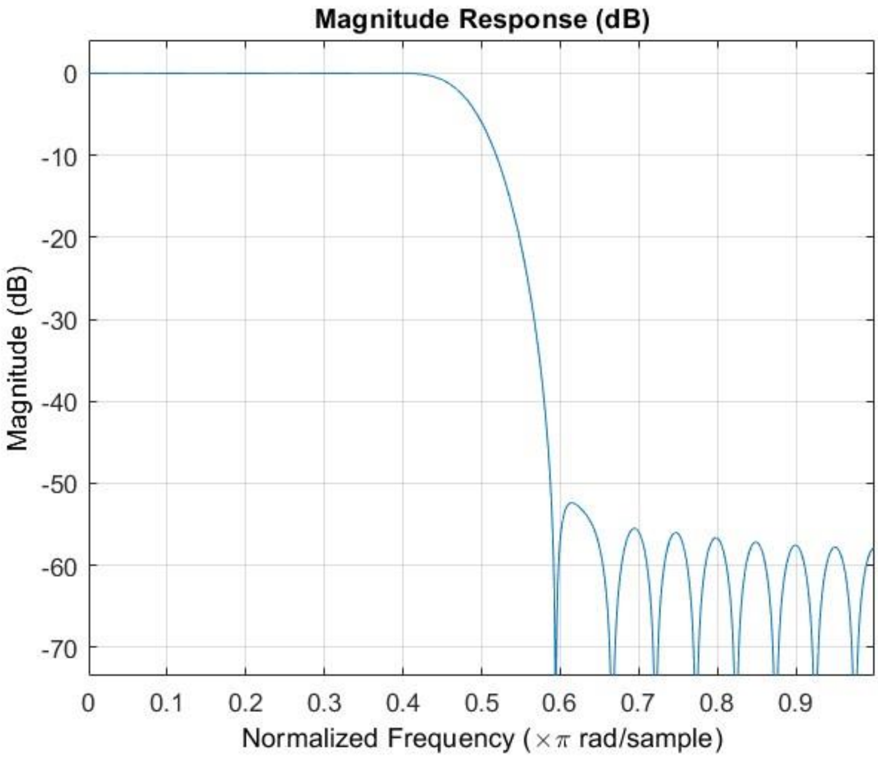
C code:

```
#include <stdio.h>
#include <math.h>

#define N 39
#define M_PI 3.14159265358979323846
void lowpass(double fc, double fs, double h_n[]){
    double wc = 2*M_PI*fc/fs;
    double h_d[N];
    double w_h[N];
    for (int n = -(N-1)/2;n<=(N-1)/2;n++){
        if (n==0){
            h_d[n+(N-1)/2]= wc/M_PI;
        }
        else {
            h_d[n+(N-1)/2] = sin (wc*n)/(M_PI *n);
        }
    }
    for (int k = 0;k<N;k++){
        w_h[k] = 0.54 - 0.46*cos(2*M_PI*k / (N-1));
    }
    for (int i =0;i<N;i++){
        h_n[i] = h_d[i]*w_h[i];
    }

}
int main(){
    double fc = 400;
    double wc = M_PI / 2;
    double fs = 2*M_PI*fc / wc;
    double h_n[N];
    lowpass(fc,fs,h_n);
    printf("Impulse response is ");
    for (int i = 0; i<N;i++){
        printf("%f",h_n[i]);
    }
    printf("\n");
    return 0;
}
```

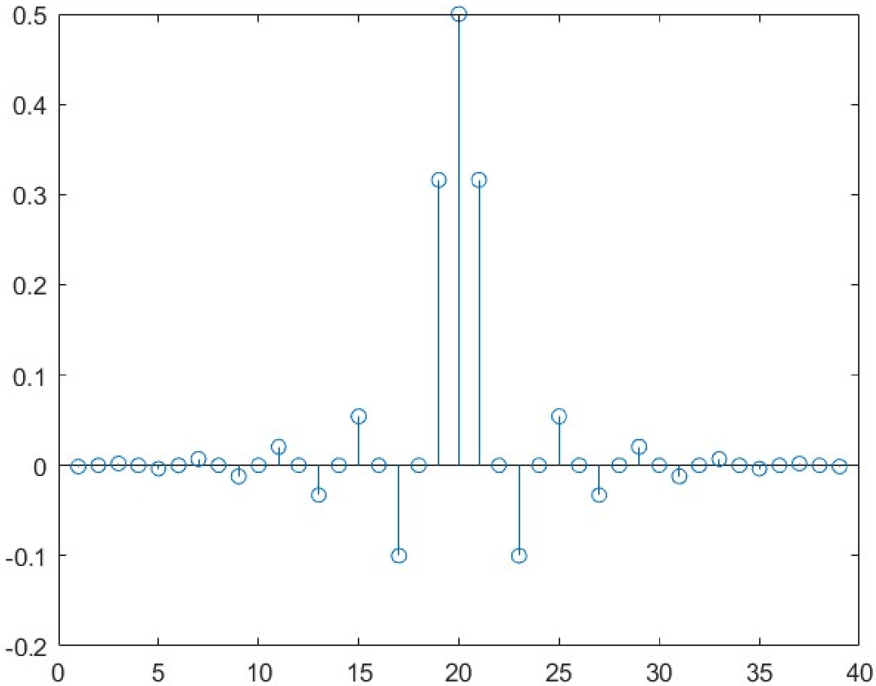
RESULT:

```
Impulse response of lpf is:
 -0.0013400.0000000.001965-0.000000-0.0037560.0000000.007062-0.000000-0.0123580.0000000.020442-
0.000000-0.0329580.0000000.054211-0.000000-0.1002210.0000000.3163130.5000000.3163130.000000-
0.100221-0.0000000.0542110.000000-0.032958-0.0000000.0204420.000000-0.012358-
0.0000000.0070620.000000-0.003756-0.0000000.0019650.000000-0.001340
```

MAGNITUDE RESPONSE:



Plot of h[n] for LPF:



BPF (Band Pass Filter):

Given,

h_d[n] = {Sin(ωc2n)/(π n)−Sin(ωc1 n)/(π n), − (N − 1)/2 ≤ n ≤ (N − 1)/2

{(ωc2 − ωc1)/π, n = 0

w_h[n] = {0.54 − 0.46 cos (2πn/(N−1)) , if 0 ≤ n ≤ N − 1

{0, otherwise

MATLAB code:

```
fc1= 500;
fc2=1200;
fs=6000;
h_n = bandpassfilter(fc1,fc2,fs);
disp(h_n);
stem(h_n);
mag_response = fvtool(h_n);
function h_n = bandpassfilter(fc1,fc2,fs)
    N=39;
    wc1 = 2*pi*fc1/fs;
    wc2 = 2*pi*fc2/fs;
    h_d = zeros(1,N);
    for n = -(N-1)/2:(N-1)/2
        if n == 0
            h_d(n+(N-1)/2 +1)=(wc2-wc1)/pi;
        else
            h_d(n+(N-1)/2 +1)=((sin(wc2*n))/(pi*n)-(sin(wc1*n))/(pi*n));
        end
    end
    w_h = zeros(1,N);
    for k = 0:N-1
        if k >=0 && k<=N-1
        w_h(k+1) = 0.54-0.46*cos((2*pi*k)/(N-1));
        else
            w_h(k+1)=0;
        end
    end
    h_n = h_d.*w_h;
end
```

RESULT:

```
>> bpf

 Columns 1 through 10

  -0.0006  -0.0009   0.0002   0.0002  -0.0038  -0.0094  -0.0077   0.0055   0.0179
0.0138

 Columns 11 through 20
```

0.0010 0.0072 0.0359 0.0399 -0.0271 -0.1306 -0.1591 -0.0432 0.1427
0.2333

Columns 21 through 30

0.1427 -0.0432 -0.1591 -0.1306 -0.0271 0.0399 0.0359 0.0072 0.0010
0.0138

Columns 31 through 39

0.0179 0.0055 -0.0077 -0.0094 -0.0038 0.0002 0.0002 -0.0009 -0.0006

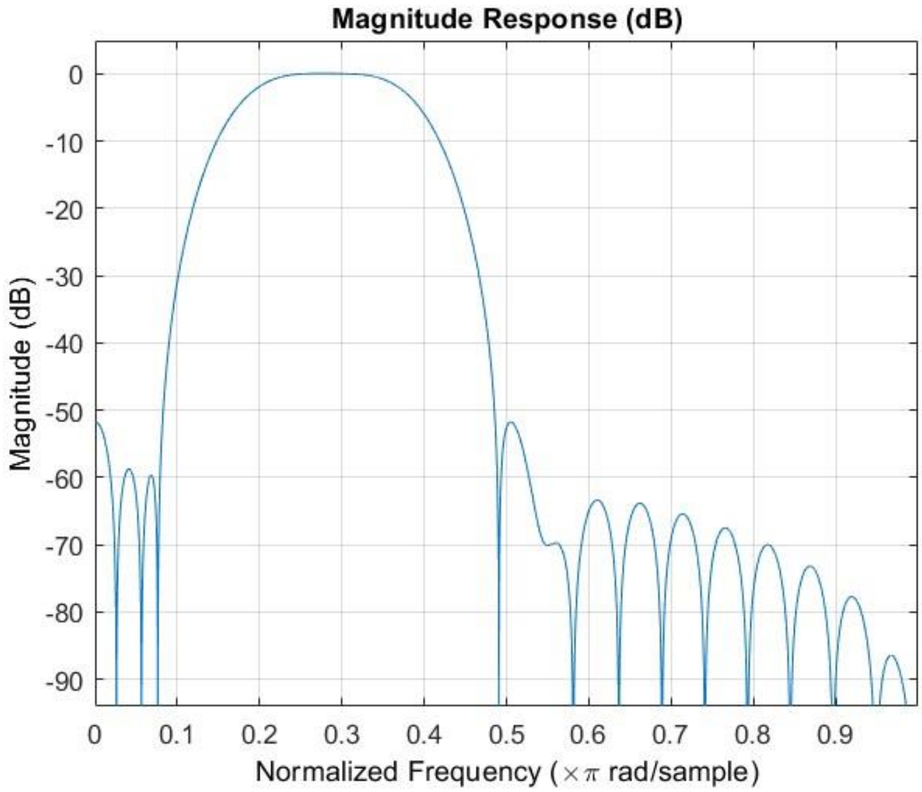
C code:

```
#include <stdio.h>
#include <math.h>

#define N 39
#define M_PI 3.14159265358979323846

void bandpass(double fc1,double fc2,double fs,double h_n[N]){
    double wc1 = 2*M_PI*fc1/fs;
    double wc2 = 2*M_PI*fc2/fs;
    double h_d[N];
    double w_h[N];
    for (int n=-(N-1)/2;n<(N-1)/2;n++){
        if(n==0){
            h_d[n+(N-1)/2]=wc2-wc1/M_PI;
        }
        else {
            h_d[n+(N-1)/2] = (sin(wc2*n)-sin(wc1*n))/(M_PI*n);
        }
    }
    for (int i = 0;i<(N);i++){
        w_h[i] = 0.54-0.46*cos(2*M_PI*i/(N-1));
    }
    for(int j=0;j<N;j++){
        h_n[j]=h_d[j]*w_h[j];
    }
}
int main(){
    double fc1 = 500;
    double fc2 = 1200;
    double fs = 6000;
    double h_n[N];
    bandpass(fc1,fc2,fs,h_n);
    printf("The impulse response of bpf is:");
    for (int i=0;i<N;i++){
        printf("%f",h_n[i]);
    }
    printf("\n");
}
```

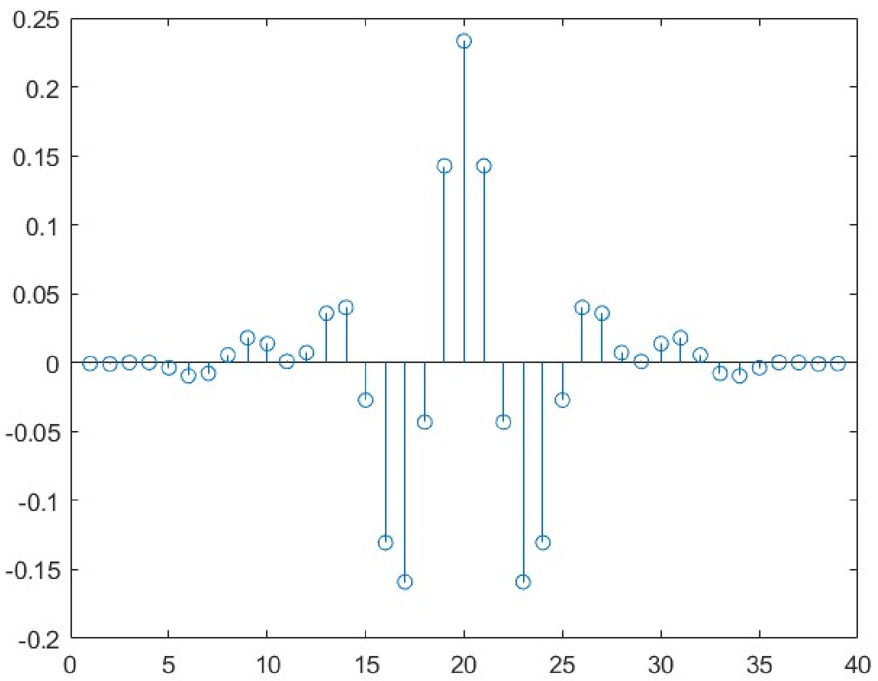
RESULT:

```
The impulse response of bpf is:
-0.000605 -0.000897 0.000172 0.000229 -0.003756 -0.009438 -0.007682 0.005538 0.017933 0.013839
0.001001 0.007228 0.035851 0.039940 -0.027106 -0.130573 -0.159129 -0.043180 0.142675 1.089970
0.142675 -0.043180 -0.159129 -0.130573 -0.027106 0.039940 0.035851 0.007228 0.001001 0.013839
0.017933 0.005538 -0.007682 -0.009438 -0.003756 0.000229 0.000172 -0.000897 0.00000
```

MAGNITUDE RESPONSE of BPF:



Plot of h[n] for BPF:



Observation:

1. **LPF:**
    - The designed LPF has a finite impulse response (FIR) with 39 taps.
    - The impulse response shows a symmetric pattern around the centre tap, as it is a linear phase FIR filter.
    - Magnitude response shows the LPF attenuates the higher frequencies components and allow the lower frequencies to pass.
    - We saw that the LPF exhibits a roll-of slope after the cutoff frequency.
2. **BPF:**
    - The designed BPF also has a finite impulse response with 39 taps.
    - The Magnitude response analysis indicates that BPF selectively passes frequencies within specified band frequencies and attenuated the rest.
    - The BPF has a peak centred around the band-pass frequency.

Conclusion:

1. **LPF:**
    - We have successfully designed the LPF design using the windowing method.
    - The LPF effectively filter out high frequency signals more than the cutoff frequency.
2. **BPF:**
    - The BPF design shows the versatility of FIR filter in implementing different frequency selective behaviour.
    - By appropriately designing the impulse response, the BPF allows only a specific band of frequencies to pass while attenuate the other.