Lab 5 – FIR Filters

Name: Shreya Mamadapur Student ID: C0774035
Instructor: Takis Zourntos

Theory: In signal processing, a finite impulse response (FIR) filter is a filter whose impulse The response is of finite length duration because it settles to zero in finite time. FIR filters have a linear phase, are stable and are typically easy to design and implement.

INTRODUCTION

In this report, we will learn how to implement FIR filters using window techniques like Hann, Hamming, Kaiser etc.

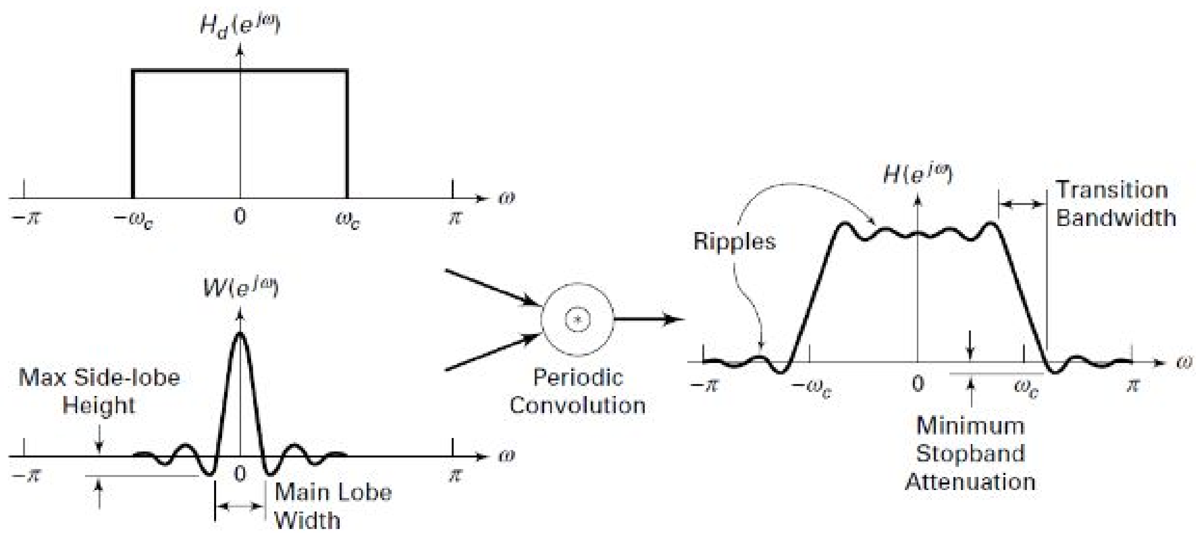
DISCUSSION

The basic idea behind the window design is to choose a proper ideal frequency-selective filter (which always has a noncausal, infinite-duration impulse response) and then to truncate (or window) its impulse response to obtain a causal linear-phase FIR filter.

Therefore the emphasis in this method is on selecting an appropriate windowing function and an appropriate ideal filter.

We will denote an ideal frequency-selective filter by Hd(ejω), which has a unity magnitude gain and linear-phase characteristics over its passband, and zero response over its stopband

To obtain an FIR filter of length M from hd(n), one has to truncate hd(n). This operation is called “windowing”. In general, h(n) can be thought of as being formed by the product of hd(n) and a window function w(n). Multiplication in time-domain is equivalent to periodic convolution in the frequency domain.



Lab 5 – FIR Filters

To limit the infinite sinc or say impulse function, we "window" it to a limited time/frequency. There are many techniques to do this. Rectangular, Hann, Hamming and Blackman methods are some popular methods. We have inbuilt functions for these in Matlab which takes an argument "M" or "L" (length of the filter). Greater the length, charger the roll-off/transition width of the filter.

Definitions of three popular windows

Hanning window:

$$w(n) = \begin{cases} 0.5\left(1 - \cos\frac{2\pi n}{M-1}\right), & \text{for } n = 0 \text{ to } N-1 \\ 0, & \text{elsewhere} \end{cases}$$

Hamming window:

$$w(n) = \begin{cases} \left(0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right)\right), & \text{for } n = 0 \text{ to } N-1 \\ 0, & \text{elsewhere} \end{cases}$$

Blackman window:

$$w(n) = \begin{cases} (0.42 - 0.5\cos\left(\frac{2\pi n}{N}\right)) + 0.08\cos\left(\frac{4\pi n}{N}\right), & \text{for } n = 0 \text{ to } N-1 \\ 0, & \text{elsewhere} \end{cases}$$

These window functions are all the "fixed window functions" meaning that for a specific M, we will have a fixed transition width.

Window Name	*Transition Width $\Delta\omega$* *Approximate*	*Exact Values*	*Min. Stopband Attenuation*
Rectangular	$\frac{4\pi}{M}$	$\frac{1.8\pi}{M}$	21 dB
Bartlett	$\frac{8\pi}{M}$	$\frac{6.1\pi}{M}$	25 dB
Hann	$\frac{8\pi}{M}$	$\frac{6.2\pi}{M}$	44 dB
Hamming	$\frac{8\pi}{M}$	$\frac{6.6\pi}{M}$	53 dB
Blackman	$\frac{12\pi}{M}$	$\frac{11\pi}{M}$	74 dB

We use these formulas in our script to get the length of the filter with the desired roll-off rate.

Lab 5 – FIR Filters

In the scripts in the Appendix, we are basically getting the desired h(n) by multiplying the appropriate ideal_filter response hd(n) with the window function w(n). A function is written for an ideal Low-pass filter and response for BPF, HPF and BSF can be easily obtained from the LPF itself. And, window functions are obtained by the built-in functions in Matlab.

CONCLUSION

Gained insights on the windowing techniques and successfully designed various FIR filters.

Lab 5 – FIR Filters

APPENDIX

1. Design a bandpass filter using the Hamming window design technique. The specifications are

Lower stopband edge: 0.3π

Upper stopband edge: 0.6π As = 50 dB

Lower passband edge: 0.4π

Upper passband edge: 0.5π Rp = 0.5 dB

Plot the impulse response and the magnitude response (in dB) of the designed filter. Do not use the fir1 function.

ANSWER

Script:

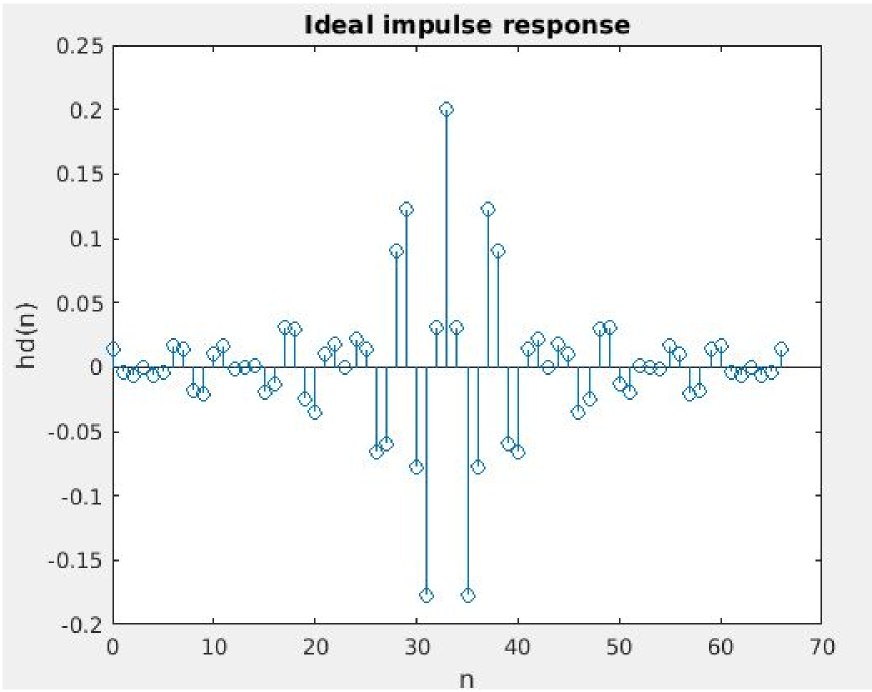
```
clc; clear; close all;

%Filter specs
ws1=0.3*pi;
wp1=0.4*pi;
wp2=0.5*pi;
ws2=0.6*pi;
Rp=0.5; Rs=50;

roll_off = min((wp1-ws1),(ws2-wp2));
wc1 = (ws1+wp1)/2; wc2 = (wp2+ws2)/2; %cut-off frequencies
M = ceil(6.6*pi/roll_off) + 1 %For hamming window mthd roll-off is 6.6*pi/M
n=[0:M-1];

hd = ideal_lp(wc2,M)-ideal_lp(wc1,M); %BPF is combination of LPF & HPF(reverse LPF)
w_ham = (hamming(M)');
h = hd .* w_ham;
[db,mag,pha,w] = freqz_m(h,[1]); delta_w = pi/500;

Rpd = -min(db((wp1/delta_w)+1:(wp2/delta_w)+1)), % Actual passband ripple
Asd = floor(-max(db(1:(ws1/delta_w)+1))), % Actual Attn


figure; stem(n,w_ham); title('Hamming window');
xlabel('n'); ylabel('w(n)');

figure; stem(n,hd); title('Ideal impulse response');
xlabel('n'); ylabel('hd(n)');
figure; stem(n,h); title('Hamming filtered impulse response');
xlabel('n'); ylabel('h(n)');
```

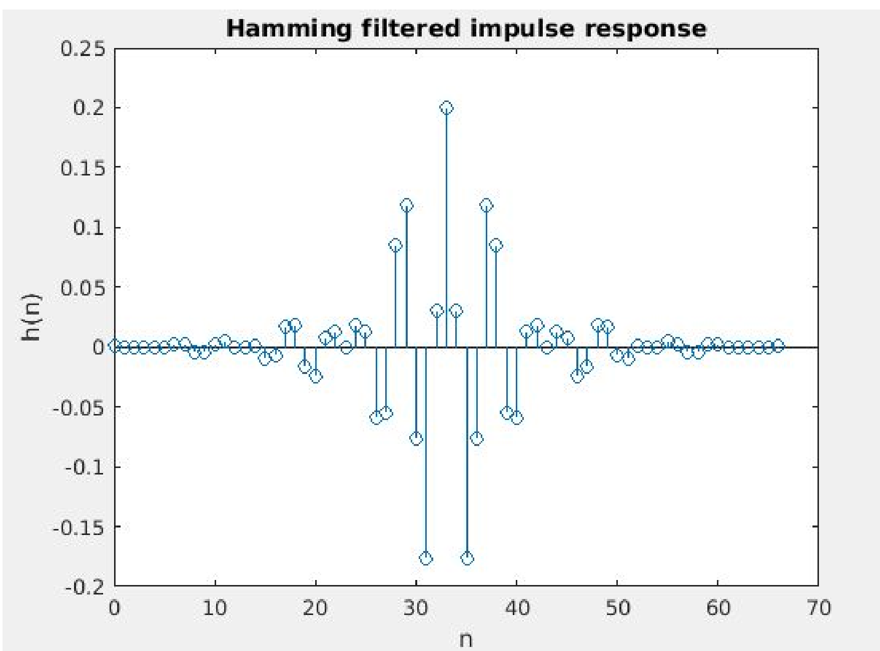
Lab 5 – FIR Filters

```
figure; plot(w/pi,db,'linewidth',1); title('Magnitude response in dB');
xlabel('Frequency (w/pi)'); ylabel('deciBels');


function hd = ideal_lp(wc,M);
% Ideal lowpass filter computation
% --------------------------------
% [hd] = ideal_lp(wc,M)
% hd = ideal impulse response between 0 to M-1
% wc = cutoff frequency in radians
%M = length of the ideal filter
%
alpha = (M-1)/2; n = [0:1:(M-1)];
m = n - alpha; fc = wc/pi; hd = fc*sinc(fc*m);
end


function [db,mag,pha,w] = freqz_m(b,a);
% Modified version of freqz subroutine
% ------------------------------------
% [db,mag,pha,w] = freqz_m(b,a);
% db = relative magnitude in dB computed over 0 to pi radians
% mag = absolute magnitude computed over 0 to pi radians
% pha = phase response in radians over 0 to pi radians
%w = 501 frequency samples between 0 to pi radians
%b = numerator polynomial of H(z)(for FIR: b=h)
%a = denominator polynomial of H(z) (for FIR: a=[1])

[H,w] = freqz(b,a,1000,'whole');
H = (H(1:1:501))'; w = (w(1:1:501))';
mag = abs(H); db = 20*log10((mag+eps)/max(mag));
pha = angle(H);

figure; plot(mag); title('Magnitude response');
xlabel('Frequency in Hz'); ylabel('Magnitude');

end
```

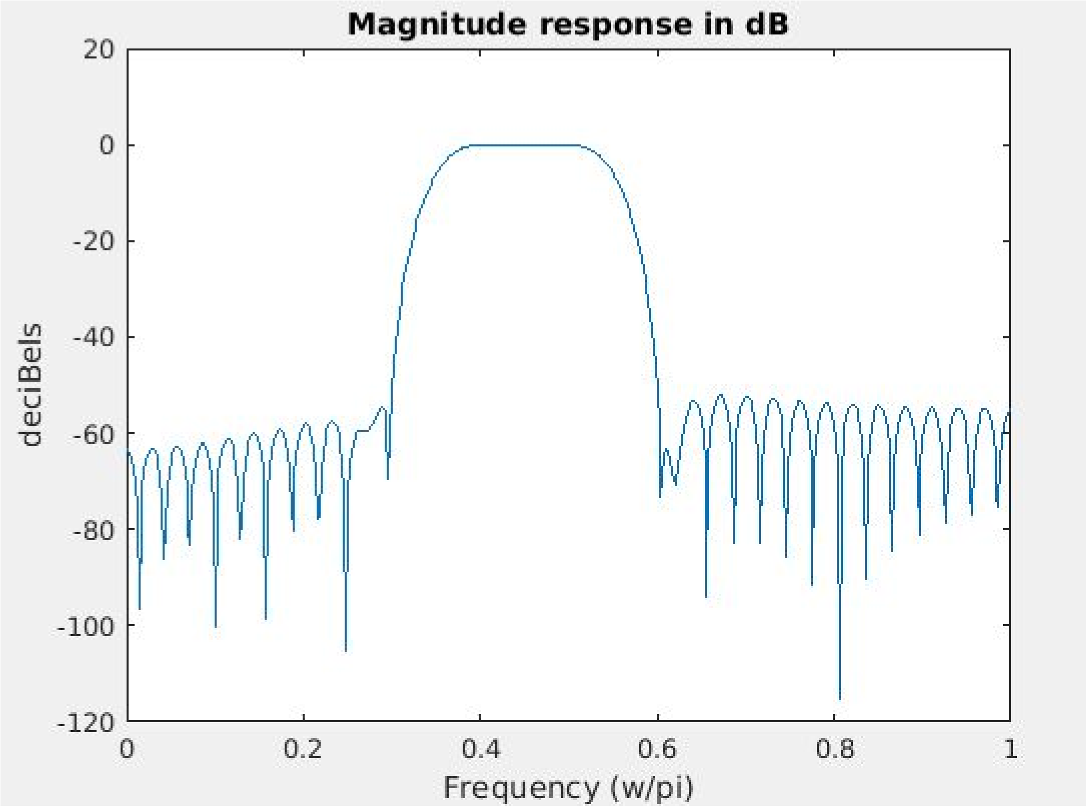
Lab 5 – FIR Filters

Result:





Lab 5 – FIR Filters



2. Design a highpass filter using one of the fixed window functions. The specifications are
Stopband edge: 0.4π, As = 50 dB
Passband edge: 0.6π, Rp = 0.004 dB
Plot the zoomed magnitude response (in dB) of the designed filter in the passband to verify the passband ripple Rp. Do not use the fir1 function.
ANSWER

Script:

```
clc; clear; close all;
%filter specs
ws = 0.4*pi; wp = 0.6*pi; roll_off = wp - ws;
As= 50; Rp=0.004;

M = ceil(11*pi/roll_off) + 1 %using blackman window technique
n=[0:M-1];
wc = (ws+wp)/2, % Ideal filter cutoff frequency
hd = ideal_lp(pi,M)-ideal_lp(wc,M);
```

Lab 5 – FIR Filters

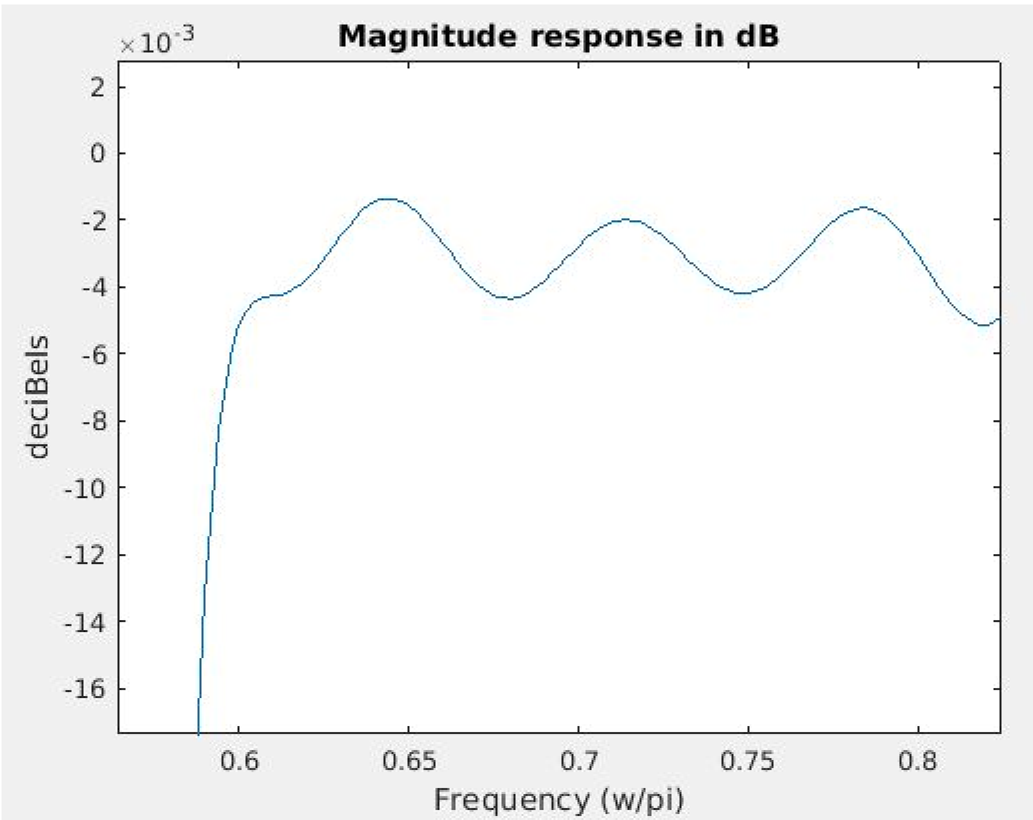
```
w_bman = (blackman(M))';
h = hd.*w_bman;
[db,mag,pha,w] = freqz_m(h,[1]); delta_w = pi/500;

Rp = -(min(db(1:1:wp/delta_w+1))); % Actual passband ripple
As = -round(max(db(ws/delta_w+1:1:501))) % Min stopband attenuation

figure; stem(n,w_bman); title('Blackman window');
xlabel('n'); ylabel('w(n)');

figure; stem(n,hd); title('Ideal impulse response');
xlabel('n'); ylabel('hd(n)');
figure; stem(n,h); title('Blackman window filtered impulse response');
xlabel('n'); ylabel('h(n)');

figure; plot(w/pi,db,'linewidth',1); title('Magnitude response in dB');
xlabel('Frequency (w/pi)'); ylabel('deciBels');

function hd = ideal_lp(wc,M);
% Ideal lowpass filter computation
% --------------------------------
% [hd] = ideal_lp(wc,M)
% hd = ideal impulse response between 0 to M-1
% wc = cutoff frequency in radians
%M = length of the ideal filter
alpha = (M-1)/2; n = [0:1:(M-1)];
m = n - alpha; fc = wc/pi; hd = fc*sinc(fc*m);
end


function [db,mag,pha,w] = freqz_m(b,a);
% Modified version of freqz subroutine
% ------------------------------------
% [db,mag,pha,w] = freqz_m(b,a);
% db = relative magnitude in dB computed over 0 to pi radians
% mag = absolute magnitude computed over 0 to pi radians
% pha = phase response in radians over 0 to pi radians
% grd = group delay over 0 to pi radians
%w = 501 frequency samples between 0 to pi radians
%b = numerator polynomial of H(z)(for FIR: b=h)
%a = denominator polynomial of H(z) (for FIR: a=[1])
```

Lab 5 – FIR Filters

```
[H,w] = freqz(b,a,1000,'whole');
H = (H(1:1:501))'; w = (w(1:1:501))';
mag = abs(H); db = 20*log10((mag+eps)/max(mag));
pha = angle(H);
end
```

Result:

Magnified image to show the pass-band ripple



3. Design a linear-phase bandpass filter using the Hann window design technique. The specifications are

Lower stopband edge: 0.2π

Upper stopband edge: 0.75π As = 40 dB

Lower passband edge: 0.35π

Upper passband edge: 0.55π Rp = 0.25 dB

Plot the impulse response and the magnitude response (in dB) of the designed filter. Do not use the fir1 function.

ANSWER

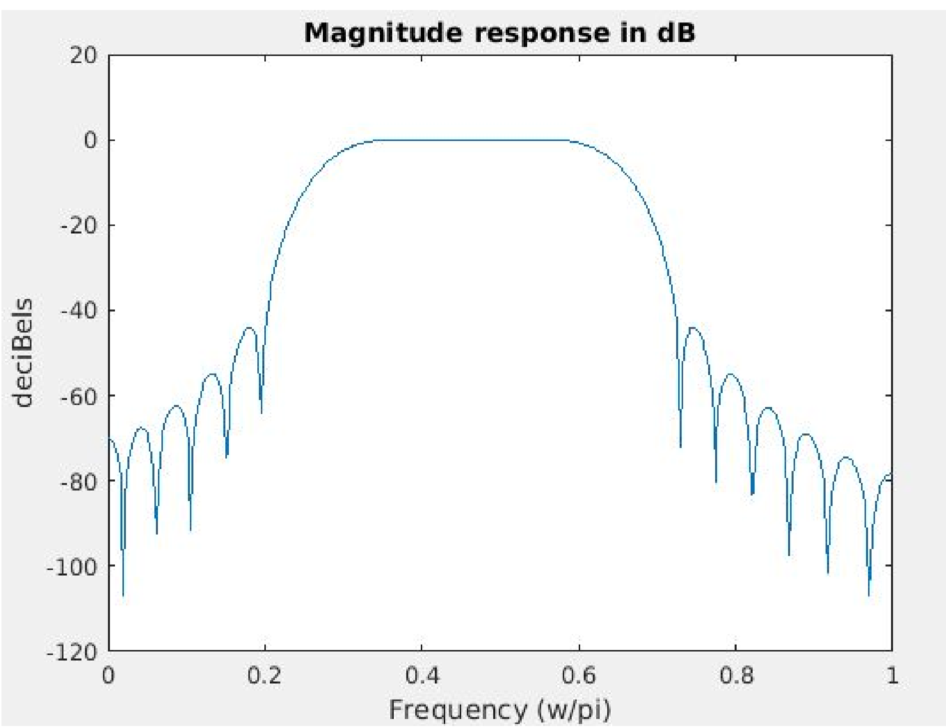
Lab 5 – FIR Filters

Script:

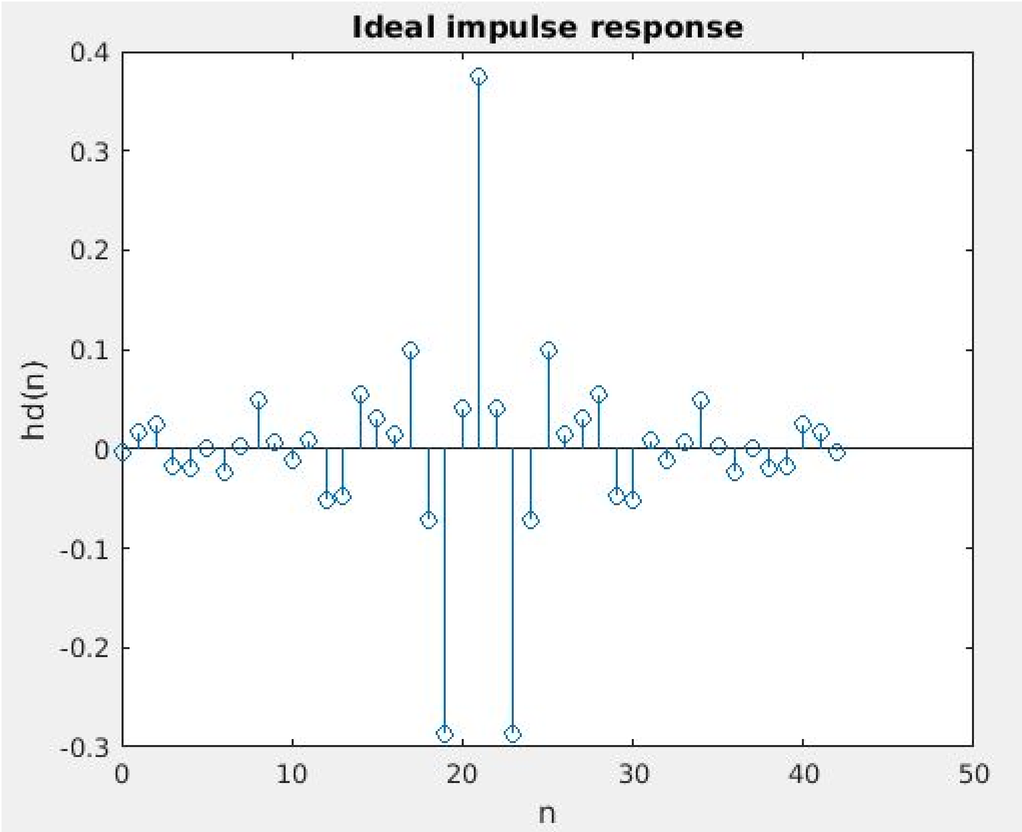
```
clc; clear; close all;

%filter specs
ws1=0.2*pi;
wp1=0.35*pi;
wp2=0.55*pi;
ws2=0.75*pi;
Rp=0.25; Rs=40;

t_width = min((wp1-ws1),(ws2-wp2)); %smallest roll-off is considered
wc1 = (ws1+wp1)/2; wc2 = (wp2+ws2)/2;
M = ceil(6.2*pi/t_width) + 1
n=[0:M-1];

hd = ideal_lp(wc2,M)-ideal_lp(wc1,M);
w_hann = (hann(M)');
h = hd .* w_hann;
[db,mag,pha,w] = freqz_m(h,[1]); delta_w = pi/500;

Rpd = -min(db((wp1/delta_w)+1:(wp2/delta_w)+1)), % Actual passband ripple
Asd = floor(-max(db(1:(ws1/delta_w)+1))), % Actual Attn

figure; stem(n,w_hann,'filled'); title('Hann window');
xlabel('n'); ylabel('w(n)');

figure; stem(n,hd); title('Ideal impulse response');
xlabel('n'); ylabel('hd(n)');
figure; stem(n,h); title('Hann filtered impulse response');
xlabel('n'); ylabel('h(n)');

figure; plot(w/pi,db,'linewidth',1); title('Magnitude response in dB');
xlabel('Frequency (w/pi)'); ylabel('deciBels');

function hd = ideal_lp(wc,M);
% Ideal lowpass filter computation
% --------------------------------
% [hd] = ideal_lp(wc,M)
% hd = ideal impulse response between 0 to M-1
% wc = cutoff frequency in radians
%M = length of the ideal filter
%
```

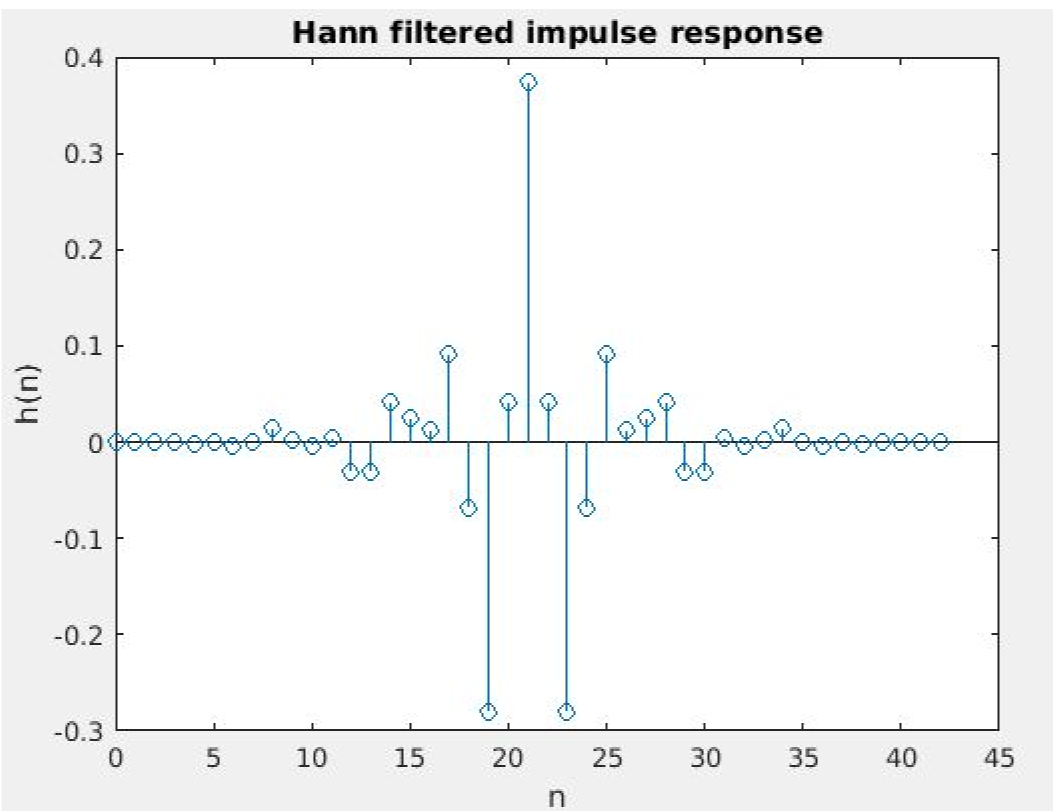
```
alpha = (M-1)/2; n = [0:1:(M-1)];
m = n - alpha; fc = wc/pi; hd = fc*sinc(fc*m);
end


function [db,mag,pha,w] = freqz_m(b,a);
% Modified version of freqz subroutine
% ------------------------------------
% [db,mag,pha,grd,w] = freqz_m(b,a);
% db = relative magnitude in dB computed over 0 to pi radians
% mag = absolute magnitude computed over 0 to pi radians
% pha = phase response in radians over 0 to pi radians
%w = 501 frequency samples between 0 to pi radians
%b = numerator polynomial of H(z)(for FIR: b=h)
%a = denominator polynomial of H(z) (for FIR: a=[1])

[H,w] = freqz(b,a,1000,'whole');
H = (H(1:1:501))'; w = (w(1:1:501))';
mag = abs(H); db = 20*log10((mag+eps)/max(mag));
pha = angle(H);
figure; plot(mag); title('Magnitude response');
end
```

Result:



Lab 5 – FIR Filters





Lab 5 – FIR Filters

4. Design a bandstop filter using the Hamming window design technique. The specifications are

Lower stopband edge: 0.4π

Upper stopband edge: 0.6π   As = 50 dB

Lower passband edge: 0.3π

Upper passband edge: 0.7π   Rp = 0.2 dB

Plot the impulse response and the magnitude response (in dB) of the designed filter. Do not use the fir1 function.

ANSWER

Script:

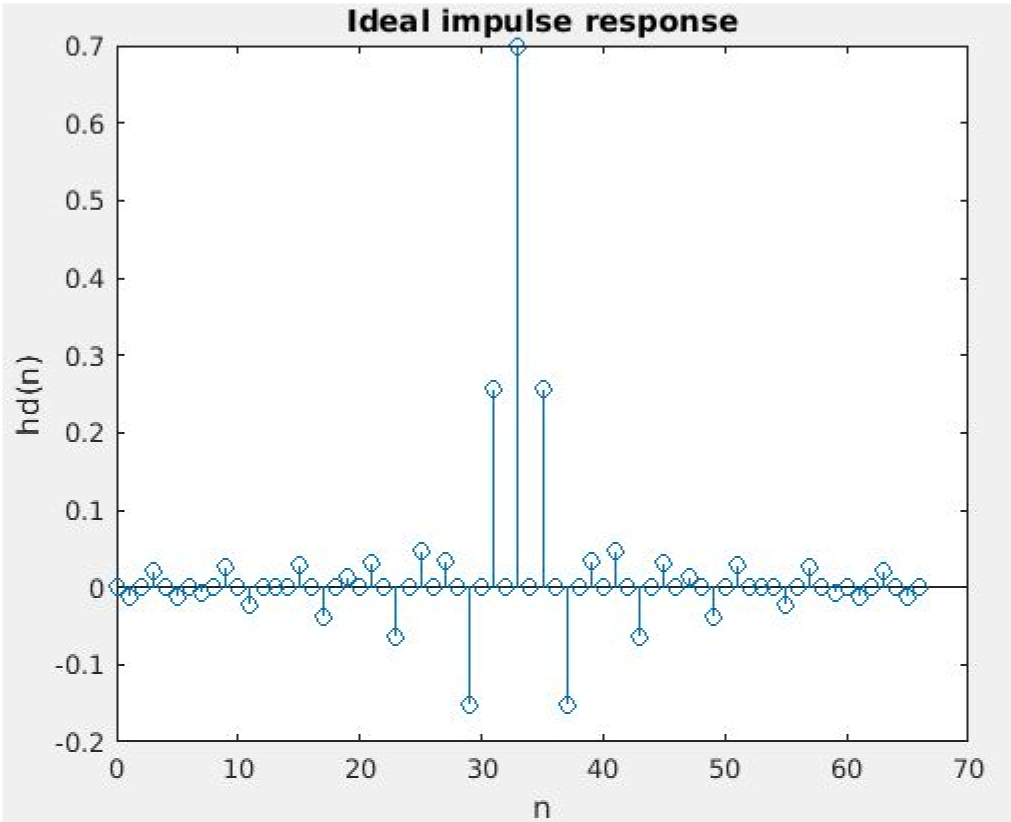
```
clc; clear; close all;

ws1=0.3*pi;
wp1=0.4*pi;
wp2=0.6*pi;
ws2=0.7*pi;
Rp=0.2; Rs=50;

roll_off = min((wp1-ws1),(ws2-wp2));
wc1 = (ws1+wp1)/2; wc2 = (wp2+ws2)/2;
M = ceil(6.6*pi/roll_off) + 1
n=[0:M-1];

hd = ideal_lp(pi,M)+ideal_lp(wc1,M)-ideal_lp(wc2,M);
w_ham = (hamming(M)');
h = hd .* w_ham;
[db,mag,pha,grd,w] = freqz_m(h,[1]); delta_w = pi/500;

Rpd = -min(db((wp1/delta_w)+1:(wp2/delta_w)+1)), % Actual passband ripple
Asd = floor(-max(db(1:(ws1/delta_w)+1))), % Actual Attn


figure; stem(n,w_ham); title('Hamming window');
xlabel('n'); ylabel('w(n)');

figure; stem(n,hd); title('Ideal impulse response');
xlabel('n'); ylabel('hd(n)');
figure; stem(n,h); title('Hamming filtered impulse response');
xlabel('n'); ylabel('h(n)');

figure; plot(w/pi,db,'linewidth',1); title('Magnitude response in dB');
```

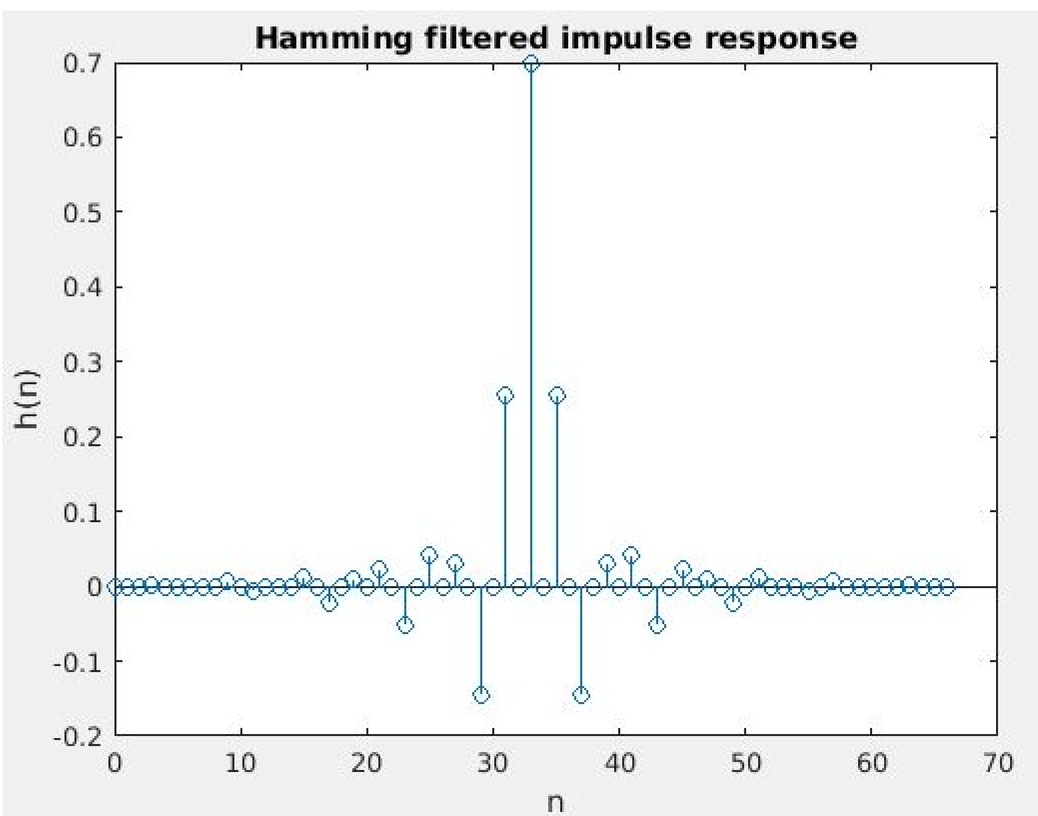
Lab 5 – FIR Filters

```
xlabel('Frequency (w/pi)'); ylabel('deciBels');


function hd = ideal_lp(wc,M);
% Ideal lowpass filter computation
% --------------------------------
% [hd] = ideal_lp(wc,M)
% hd = ideal impulse response between 0 to M-1
% wc = cutoff frequency in radians
%M = length of the ideal filter
%
alpha = (M-1)/2; n = [0:1:(M-1)];
m = n - alpha; fc = wc/pi; hd = fc*sinc(fc*m);
end


function [db,mag,pha,grd,w] = freqz_m(b,a);
% Modified version of freqz subroutine
% ------------------------------------
% [db,mag,pha,grd,w] = freqz_m(b,a);
% db = relative magnitude in dB computed over 0 to pi radians
% mag = absolute magnitude computed over 0 to pi radians
% pha = phase response in radians over 0 to pi radians
% grd = group delay over 0 to pi radians
%w = 501 frequency samples between 0 to pi radians
%b = numerator polynomial of H(z)(for FIR: b=h)
%a = denominator polynomial of H(z) (for FIR: a=[1])

[H,w] = freqz(b,a,1000,'whole');
H = (H(1:1:501))'; w = (w(1:1:501))';
mag = abs(H); db = 20*log10((mag+eps)/max(mag));
pha = angle(H); grd = grpdelay(b,a,w);

figure; plot(mag); title('Magnitude response');
xlabel('Frequency in Hz'); ylabel('Magnitude');
%figure; plot(pha); title('Phase response');
%figure; plot(grd); title('Group delay');
end
```

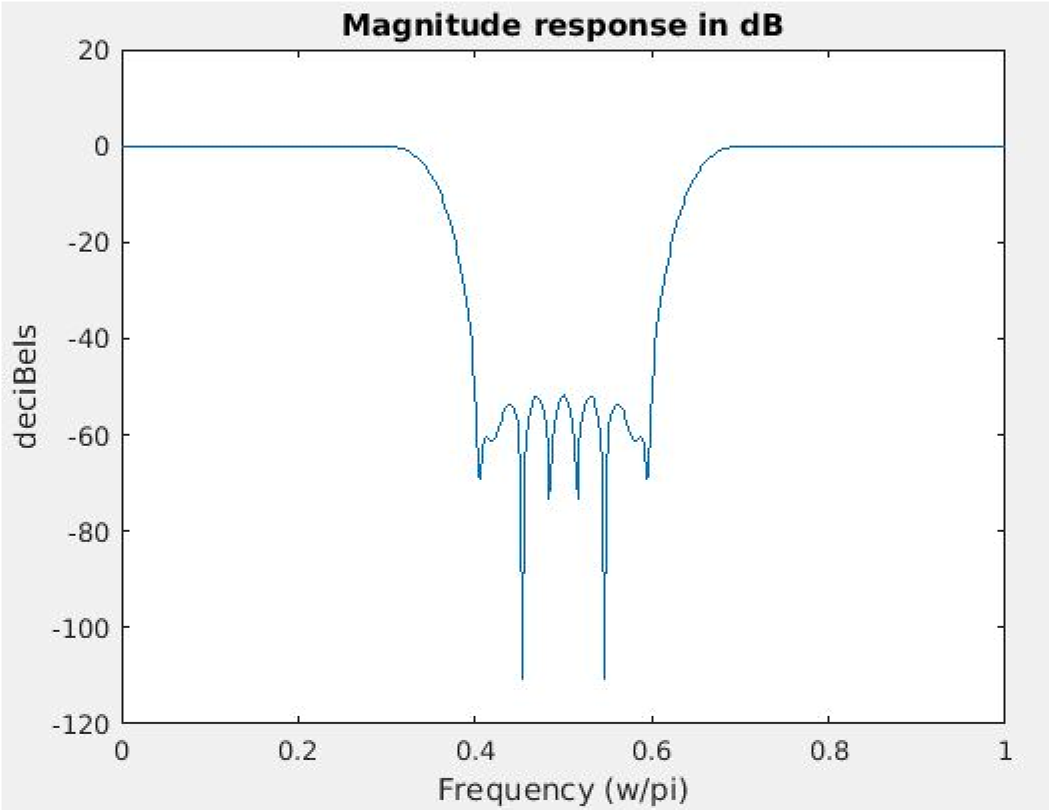
Lab 5 – FIR Filters

Result:





Lab 5 – FIR Filters



------------------------END OF DOCUMENT-----------------------