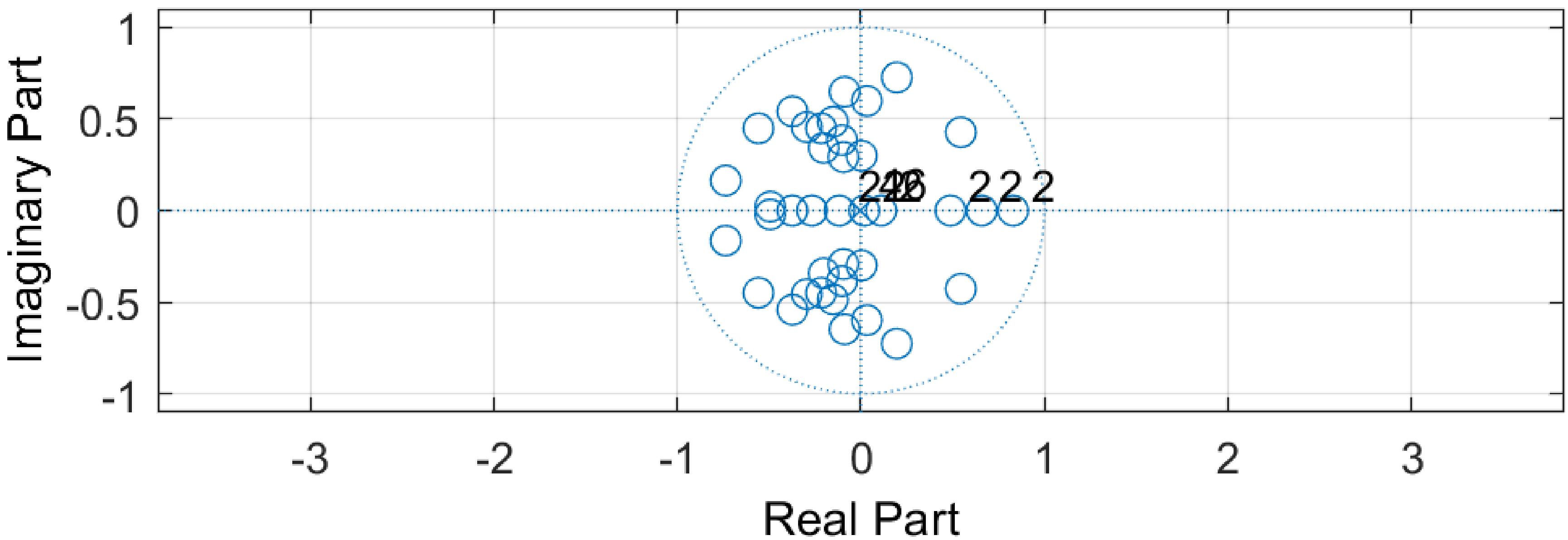
Marwin B. Alejo 2020-20221
EE274_ProgEx06

Also accessible through www.github.com/soymarwin/ee274/EE274_ProgEx06 for history tracking. Performed in MATLAB 2020b.

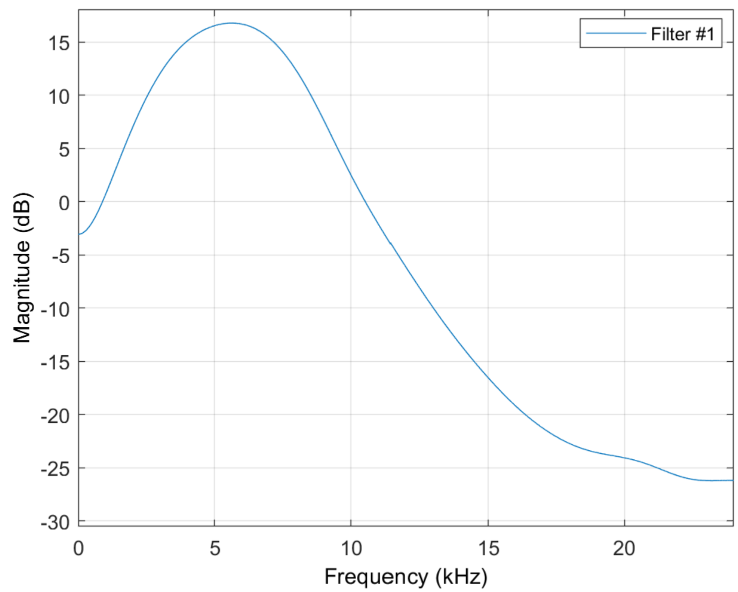
A. Pole-Zero Placement

Task in this section was performed using filter designer and was saved as an object with a filename section_A.fda. Given the magnitude response shown in the manual and through zero replacement method (no zeros required since FIR) the following below are the outputs:

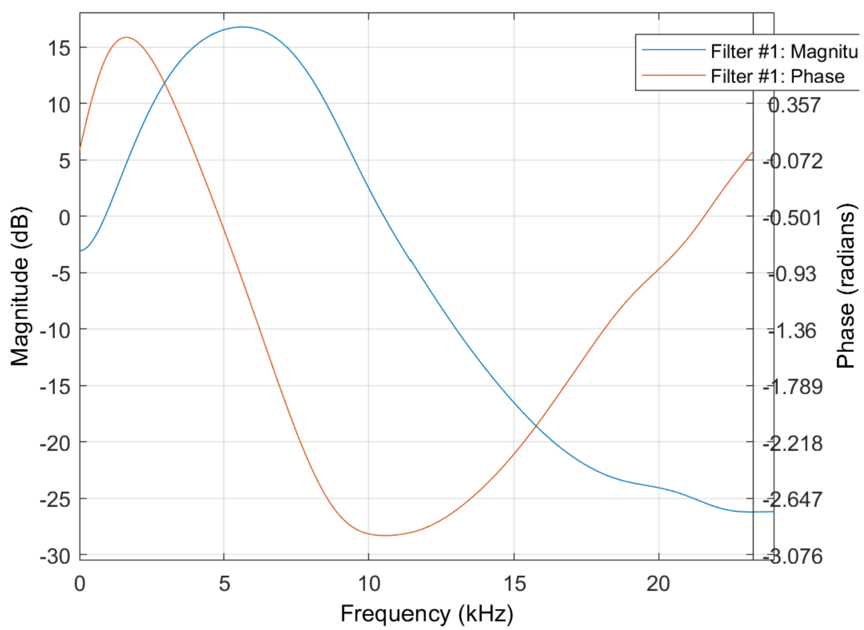
a. Zero locations of the filter. The poles are at the origin by default.

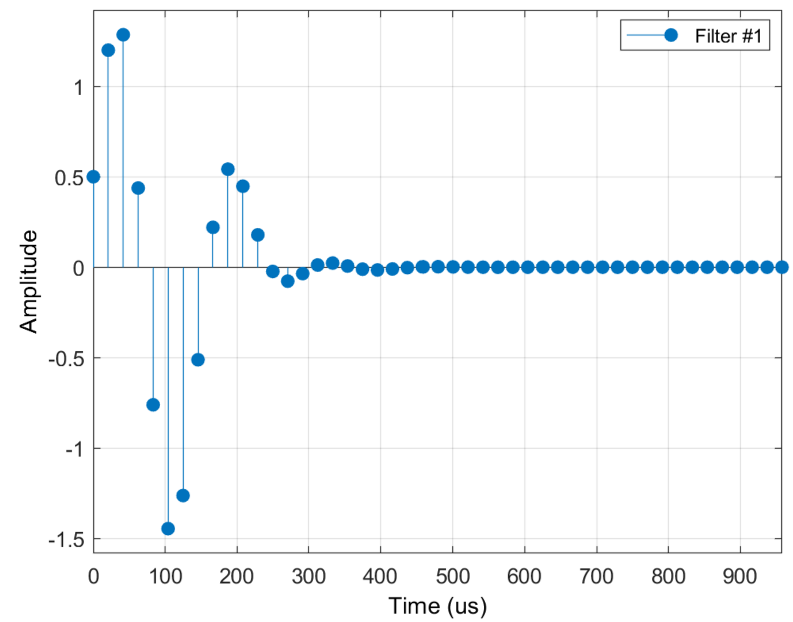


b. Magnitude Response



c. Frequency Response





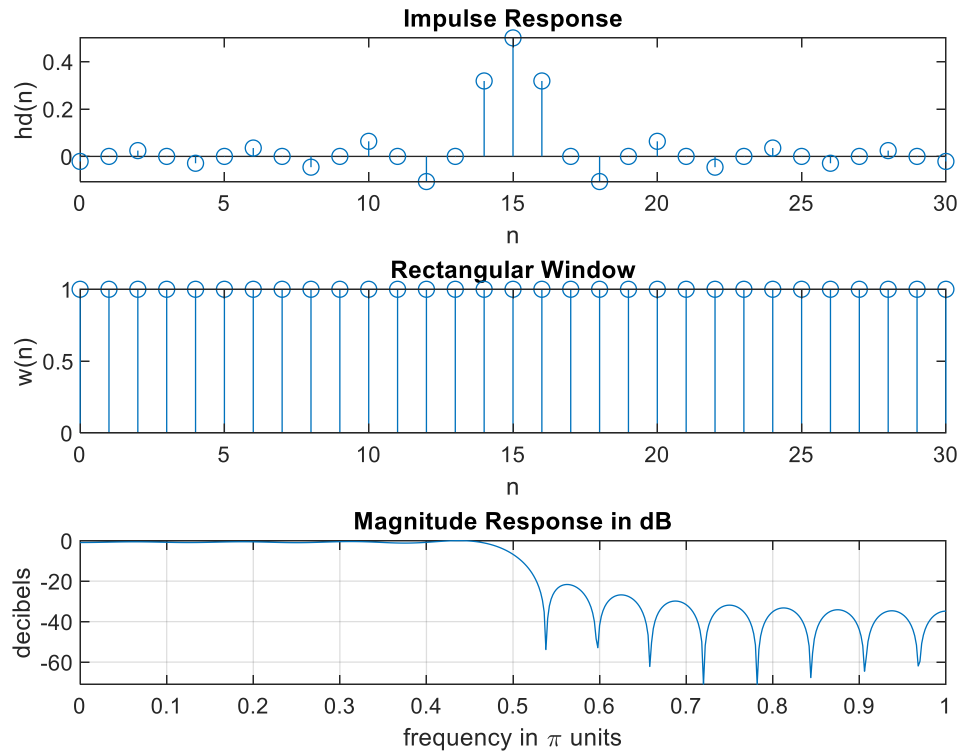
B. Properties of Various Windowing Functions

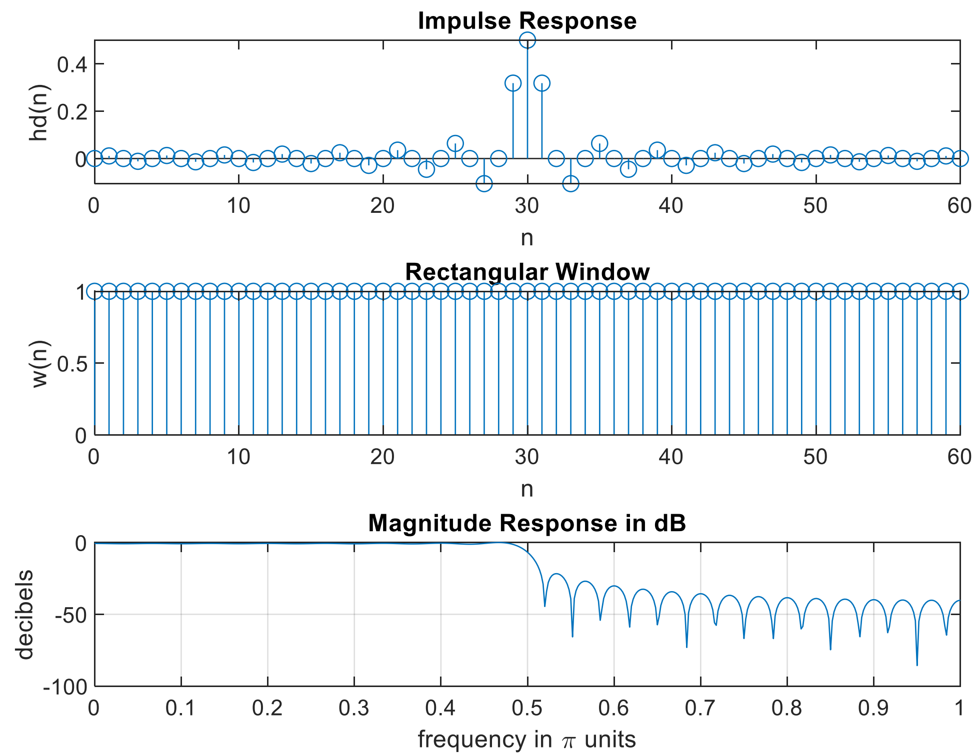
Table 1 shows the summary of the configuration for designing each FIR with respective window functions. The magnitude plot of each configuration is shown in Table 2. The MATLAB scripts used in this section are *rectw.m*, *bartw.m*, *hannw.m*, *hammw.m*, *blkw.m*, and *kaisw.m* and are shown in Table 3.

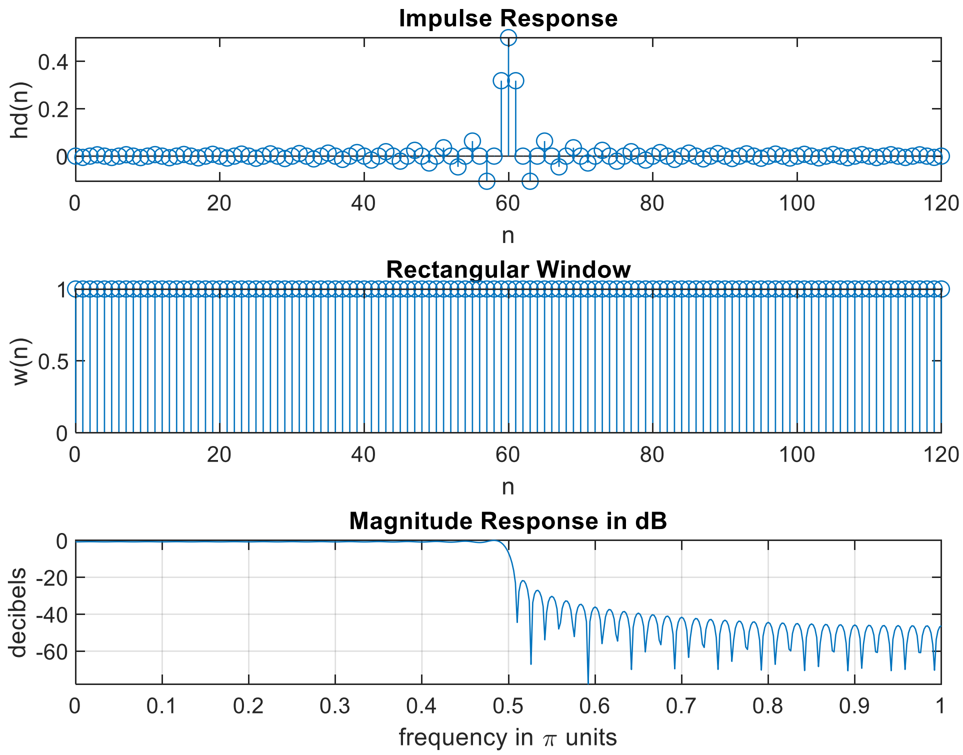
TABLE 1 FIR Design Configuration Summary

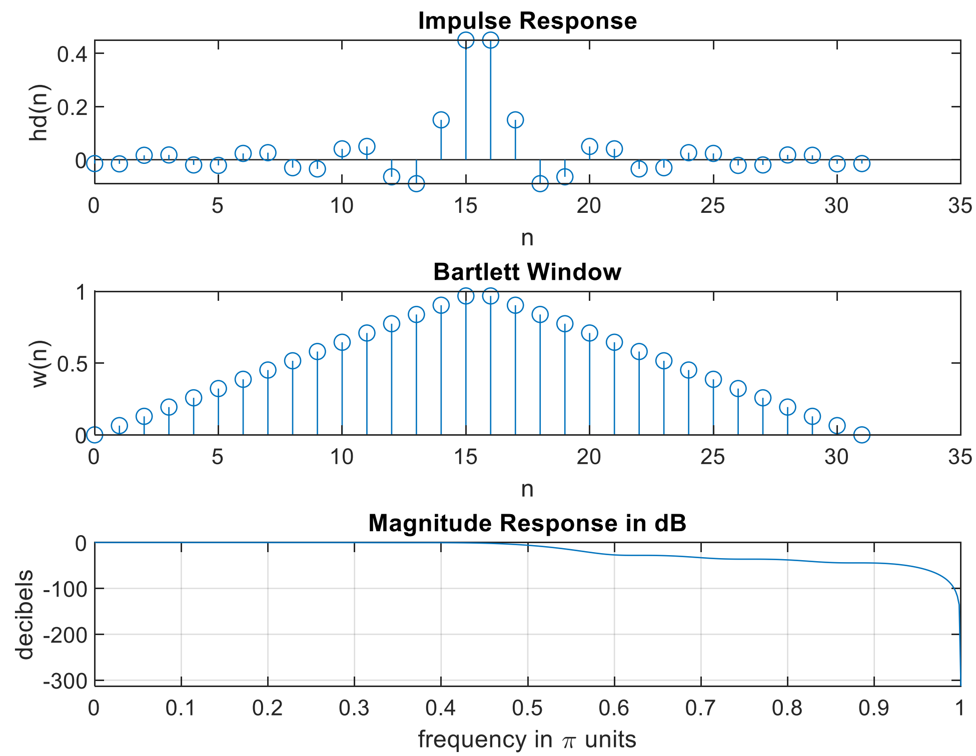
Window name	Exact Value	Transition width (in rad)			Peak sidelobe (in dB)		
		M=31	M=61	M=121	M=31	M=61	M=121
Rectangular	$\frac{1.8\pi}{M}$	0.06π	0.03π	0.015π	20.7	21	--
Bartlett	$\frac{6.1\pi}{M}$	0.203π	0.102π	0.051π	0	--	--
Hann	$\frac{6.2\pi}{M}$	0.207π	0.103π	0.052π	44	44	--
Hamming	$\frac{6.6\pi}{M}$	0.22π	0.11π	0.055π	51.4	54.3	--
Blackman	$\frac{11\pi}{M}$	0.3667π	0.183π	0.0917π	0	0	--
Kaiser Beta 1	$\frac{1.8\pi}{M}$	0.06π	0.03π	0.015π	23.4	23.3	--
Kaiser Beta 4		0.06π	0.03π	0.015π	47.6	45.4	--
Kaiser Beta 9		0.06π	0.03π	0.015π	90.7	90.3	--

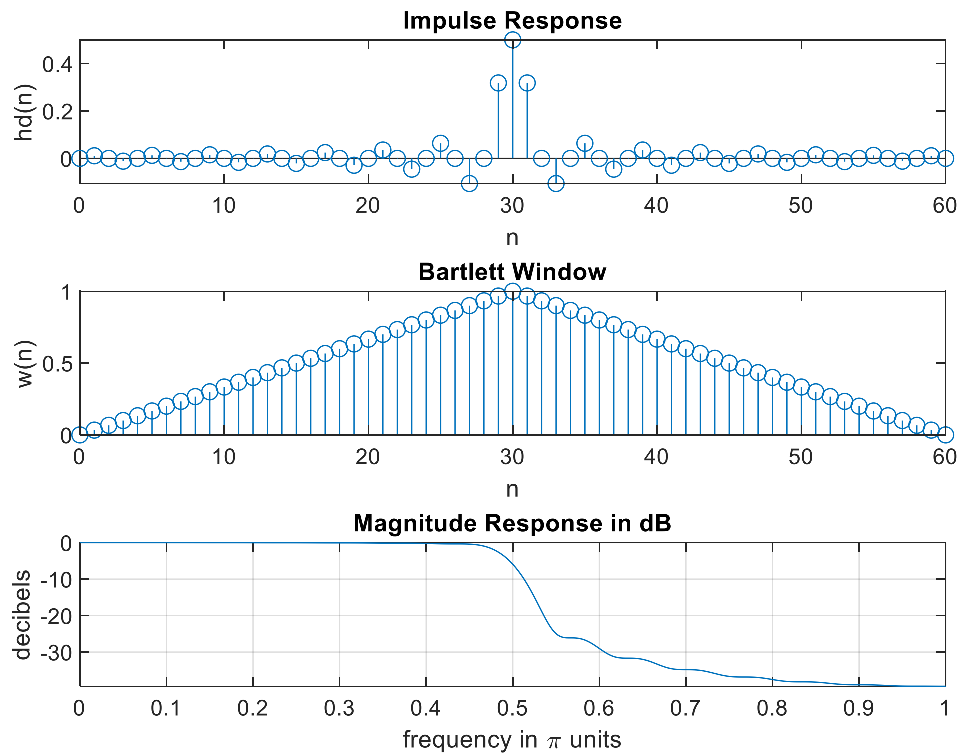
TABLE 2 Magnitude plot of each windowing configuration in Table 1

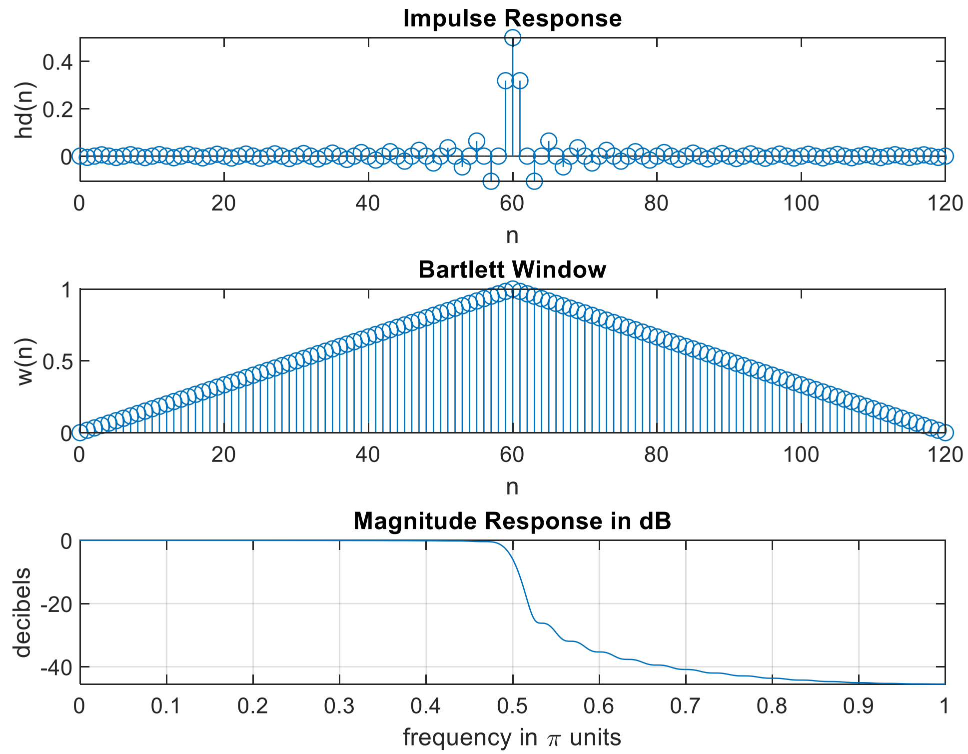
Window Name	Exact Value	Length	Plots
Rectangular	$\frac{1.8\pi}{M}$	31	

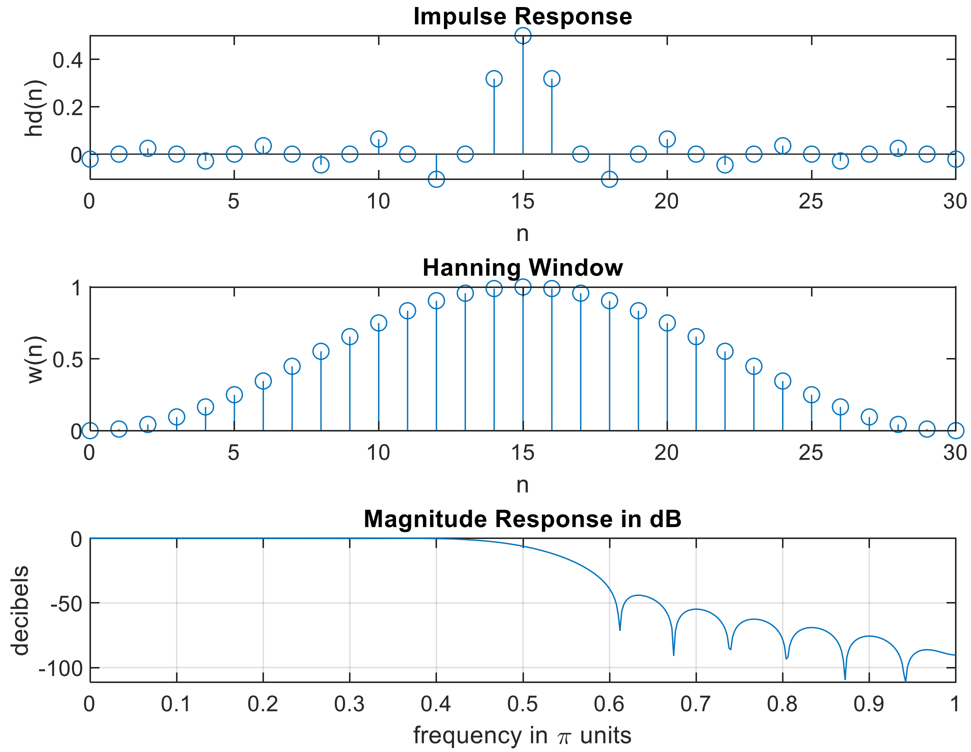
Window Name	Exact Value	Length	Plots
		61	

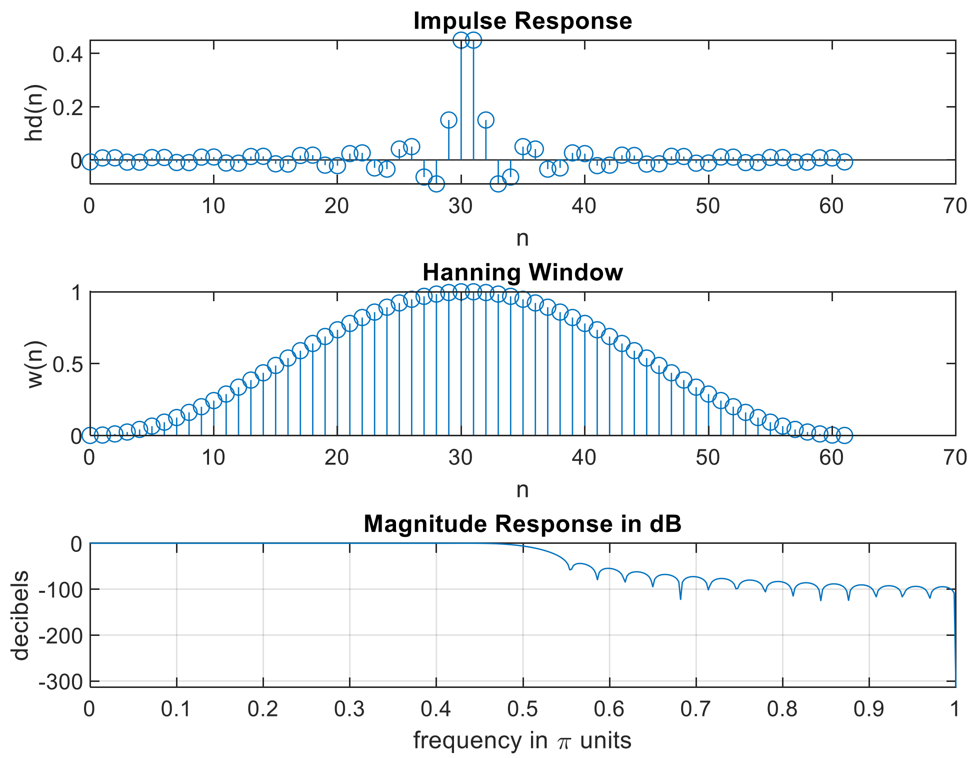
Window Name	Exact Value	Length	Plots
		121	

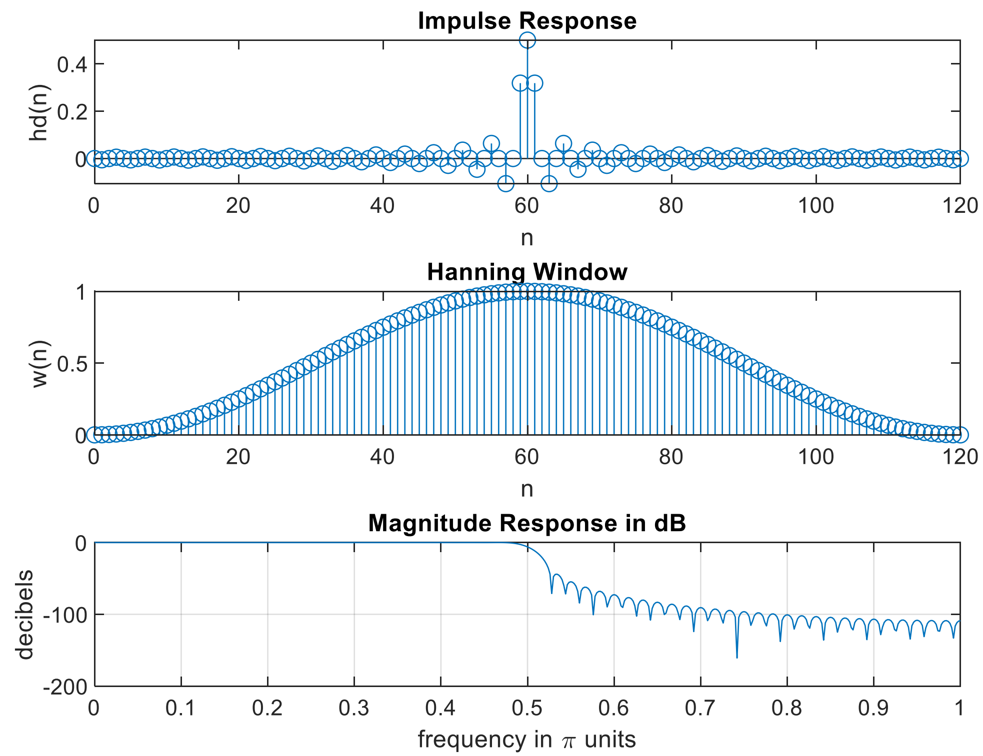
Window Name	Exact Value	Length	Plots
Bartlett	$\frac{6.1\pi}{M}$	31	

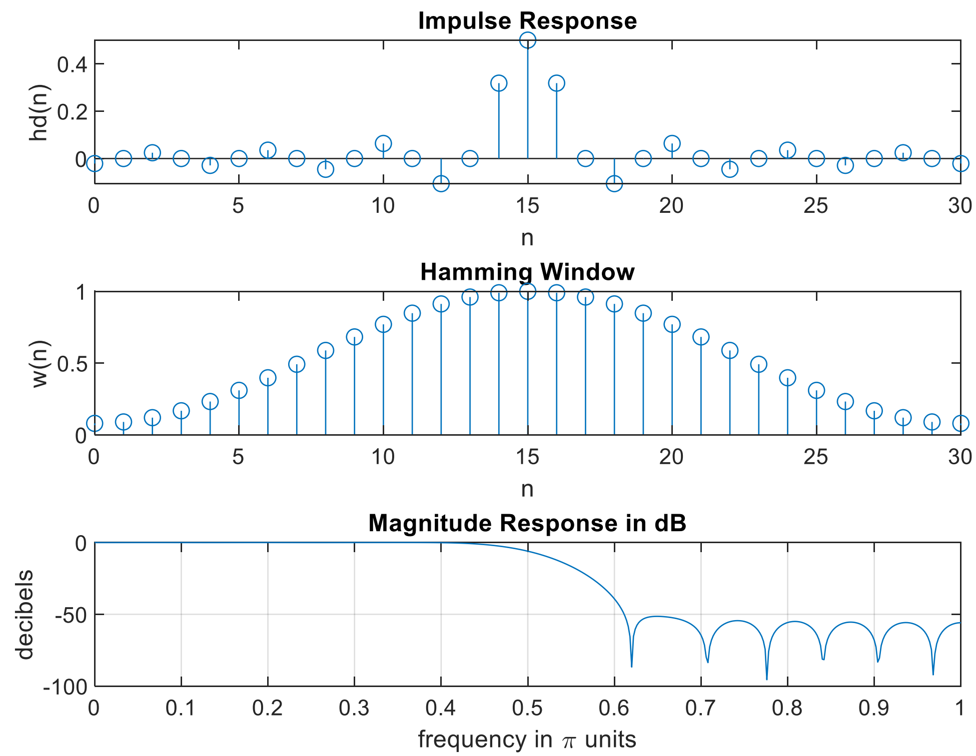
Window Name	Exact Value	Length	Plots
		61	

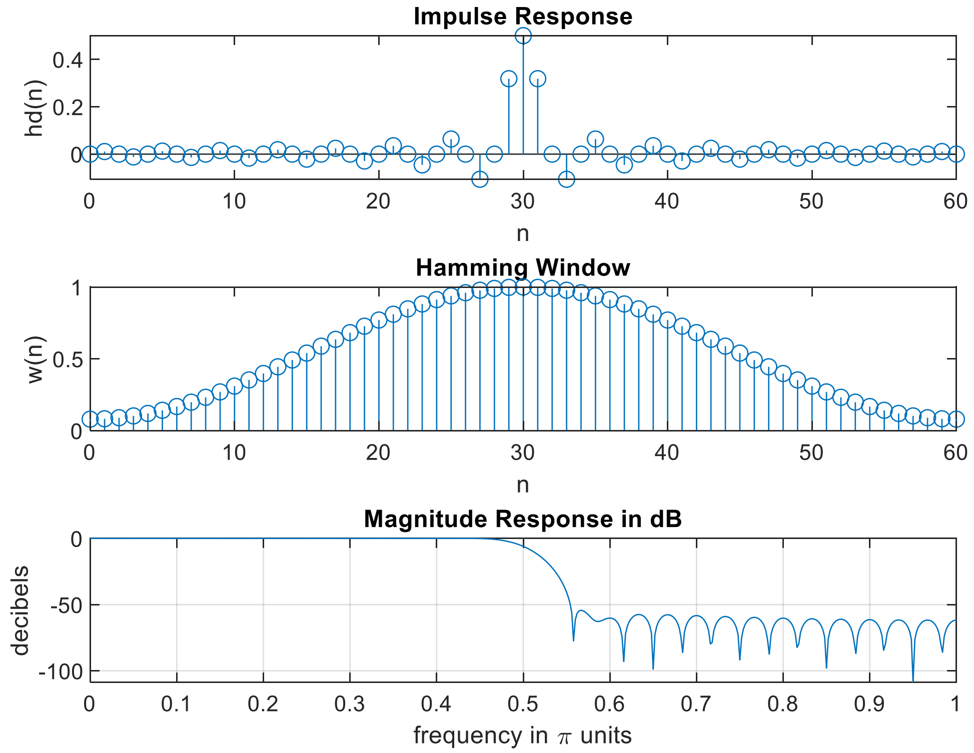
Window Name	Exact Value	Length	Plots
		121	

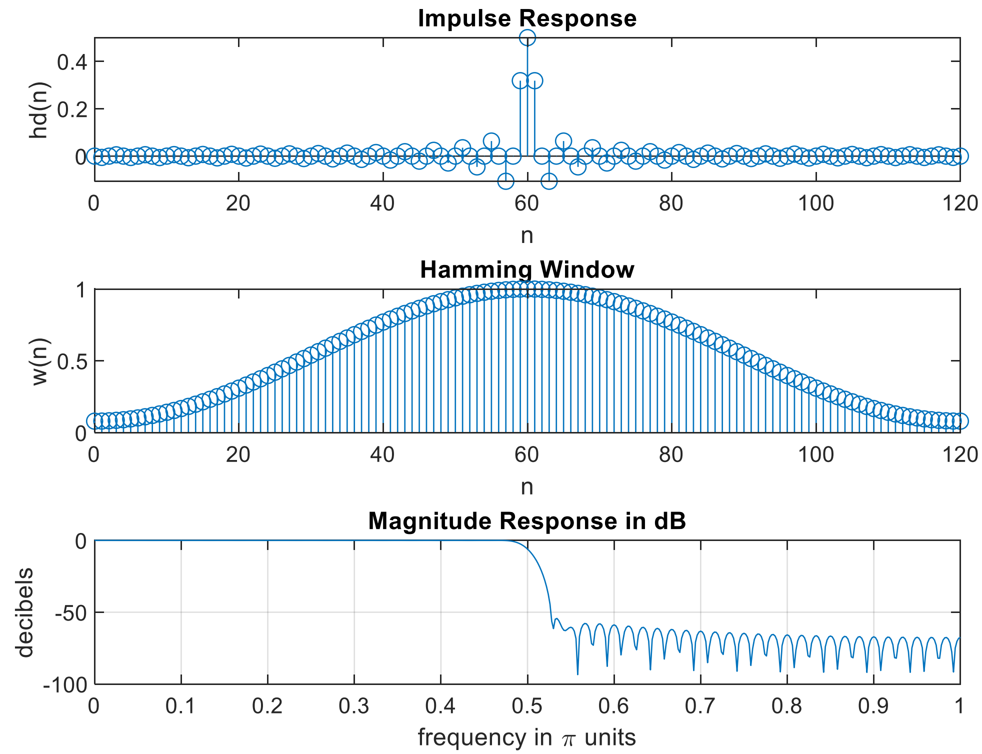
Window Name	Exact Value	Length	Plots
Hann	$\frac{6.2\pi}{M}$	31	

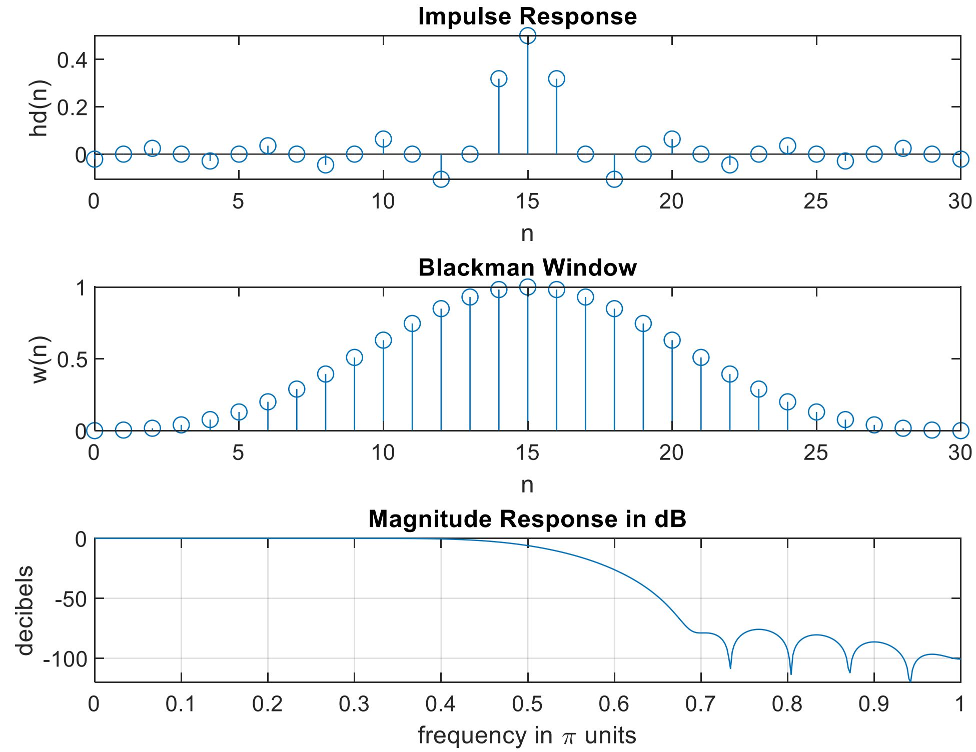
Window Name	Exact Value	Length	Plots
		61	

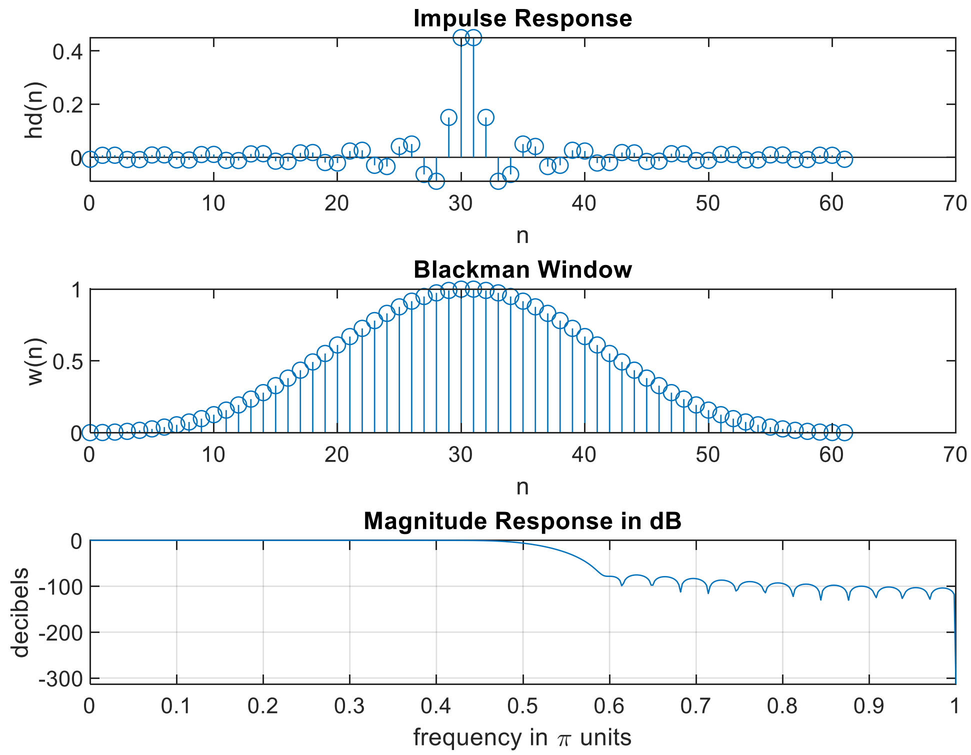
Window Name	Exact Value	Length	Plots
		121	

Window Name	Exact Value	Length	Plots
Hamming	$\frac{11\pi}{M}$	31	

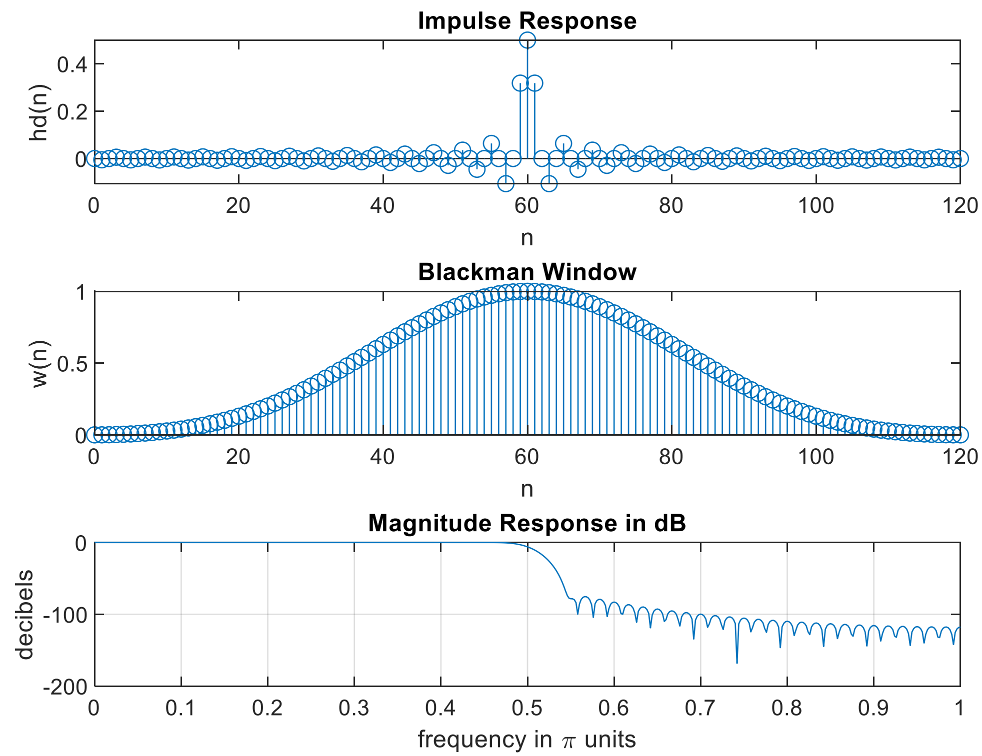
Window Name	Exact Value	Length	Plots
		61	

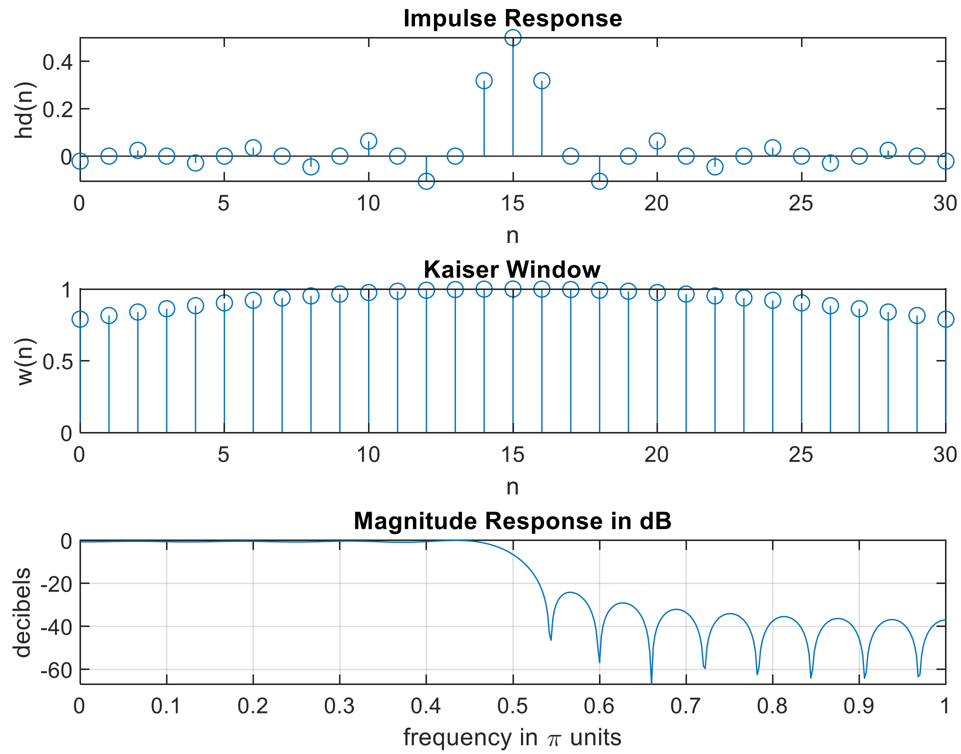
Window Name	Exact Value	Length	Plots
		121	

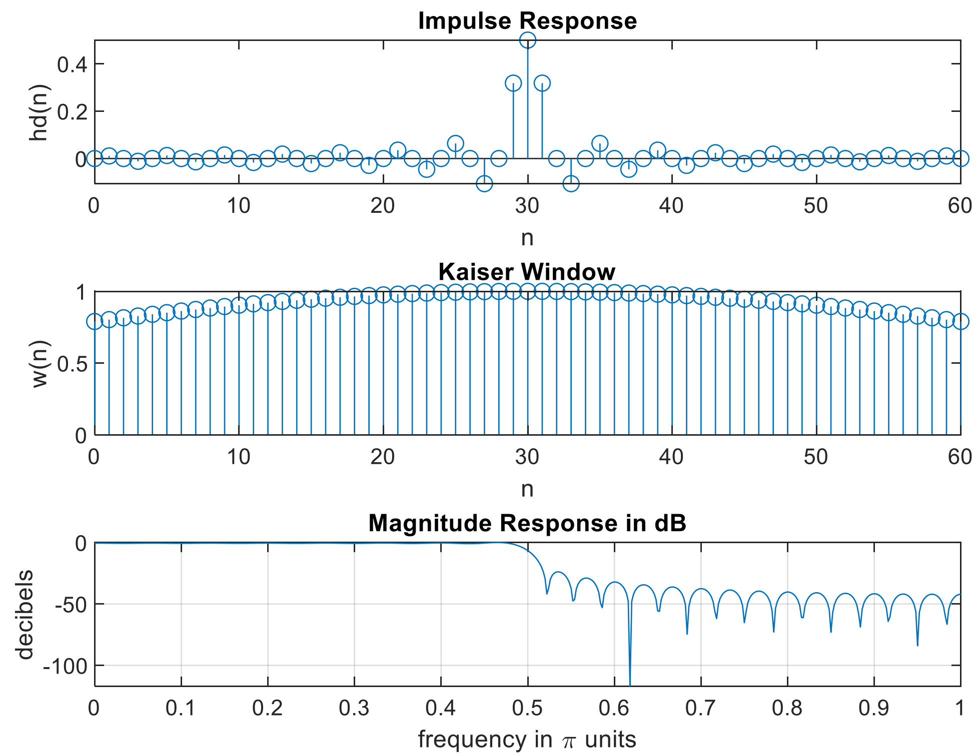
Window Name	Exact Value	Length	Plots
Blackman	$\frac{11\pi}{M}$	31	

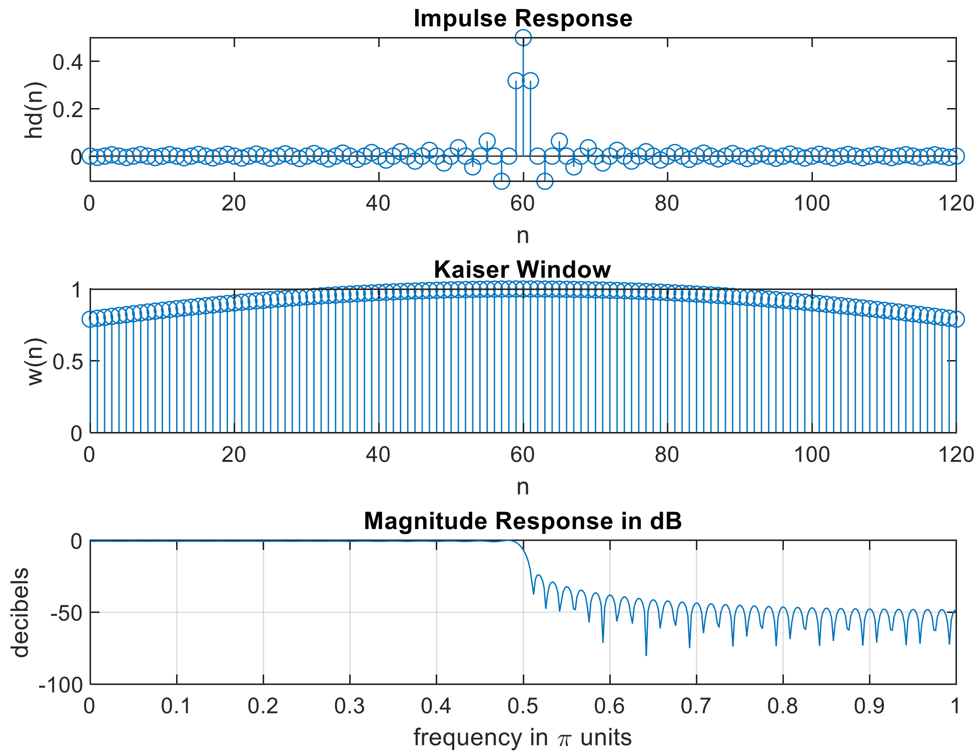
Window Name	Exact Value	Length	Plots
		61	

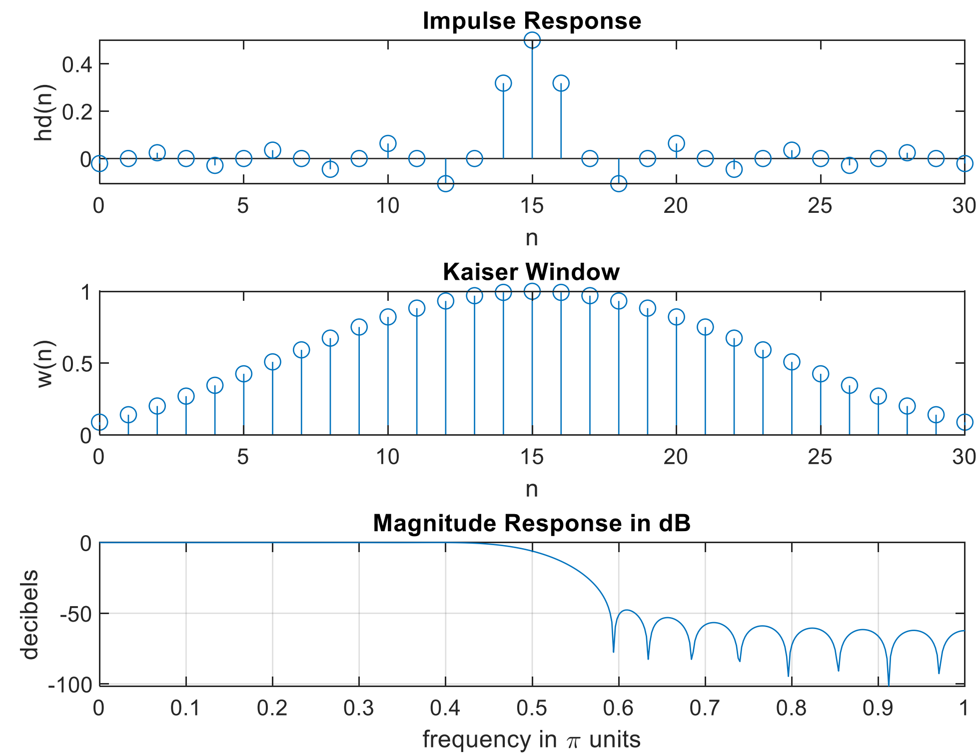
17

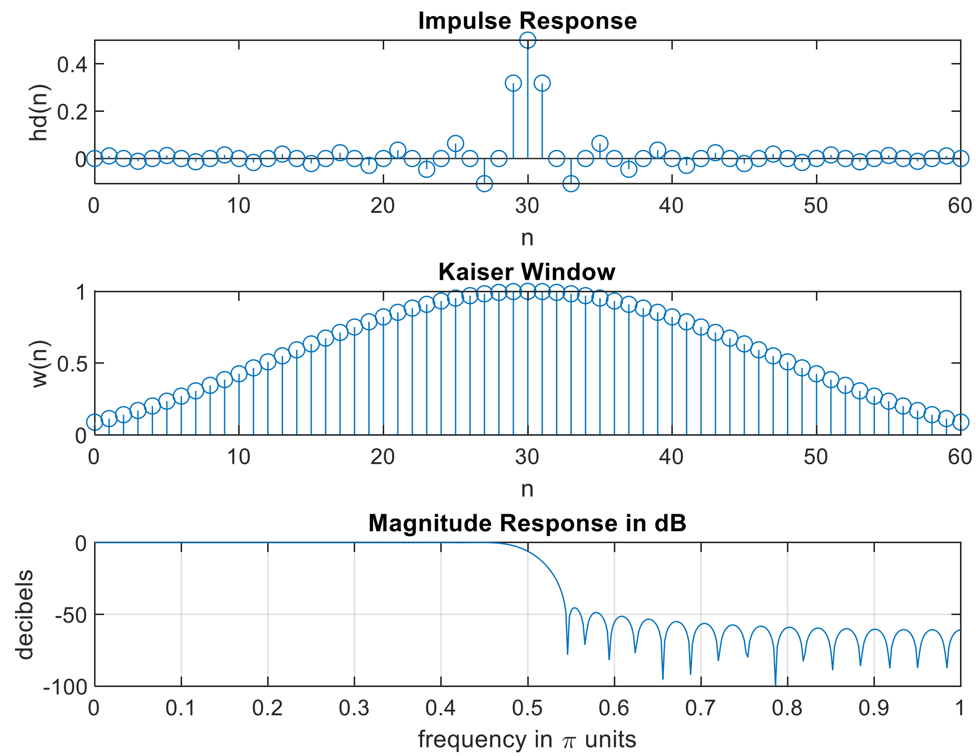
Window Name	Exact Value	Length	Plots
		121	

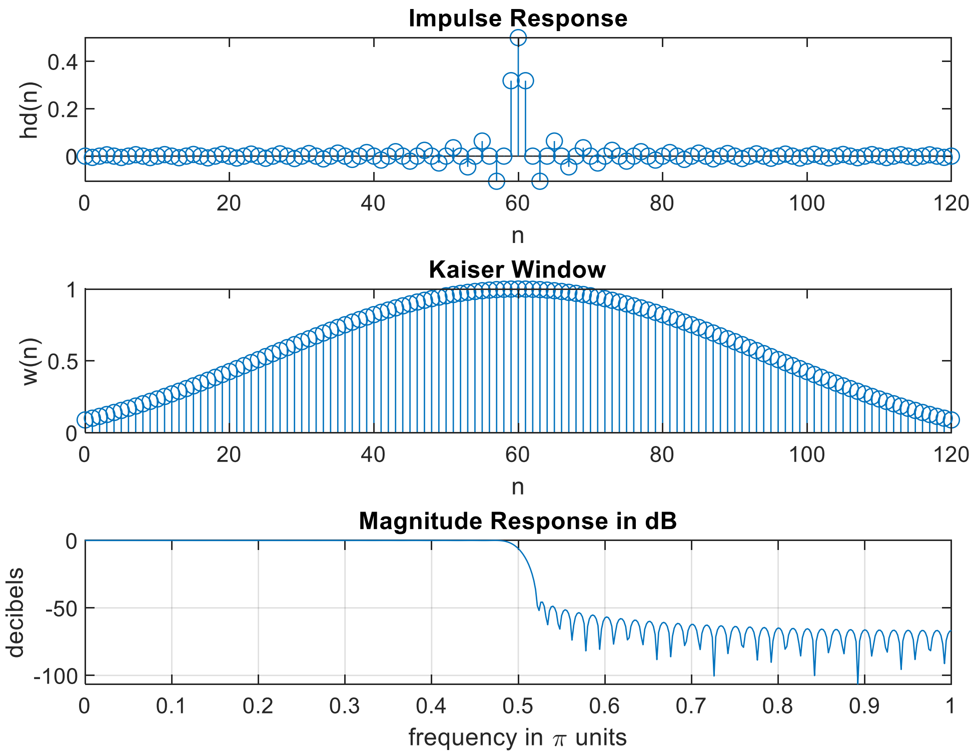
Window Name	Exact Value	Length	Plots
Kaiser Beta 1	$\frac{1.8\pi}{M}$	31	

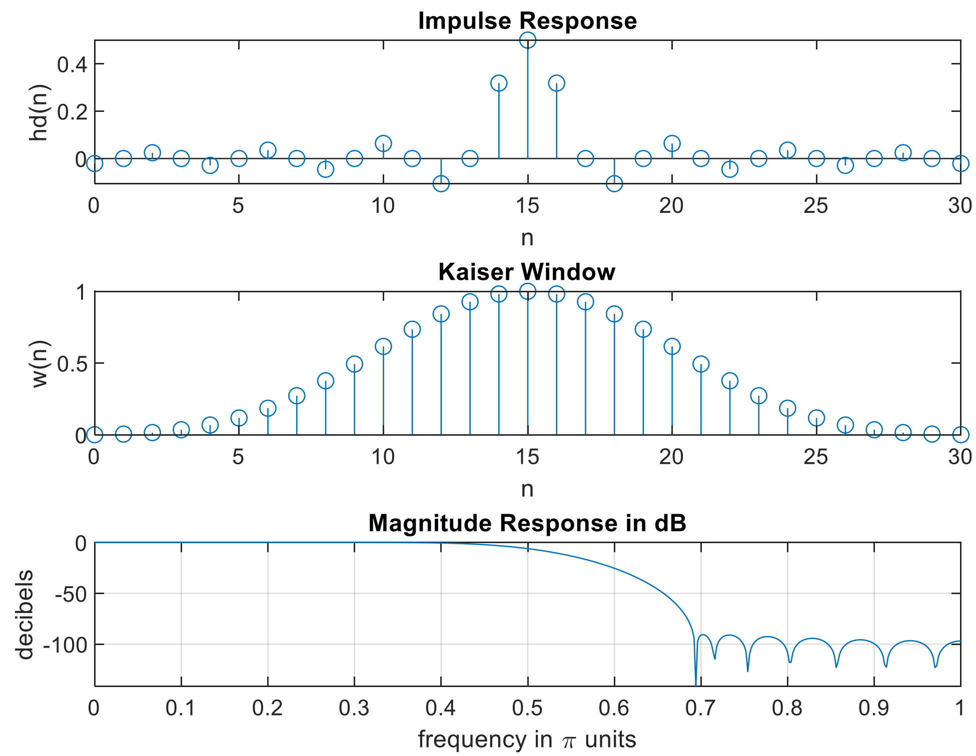
Window Name	Exact Value	Length	Plots
		61	

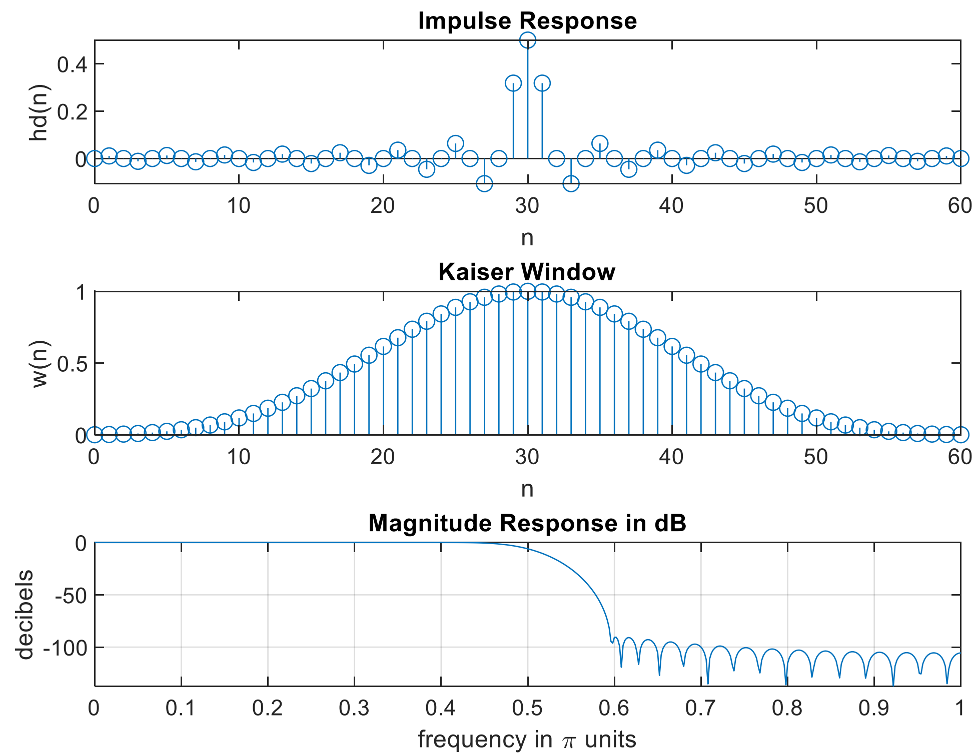
Window Name	Exact Value	Length	Plots
		121	

Window Name	Exact Value	Length	Plots
Kaiser Beta 4		31	

Window Name	Exact Value	Length	Plots
		61	

Window Name	Exact Value	Length	Plots
		121	

Window Name	Exact Value	Length	Plots
Kaiser Beta 9		31	

Window Name	Exact Value	Length	Plots
		61	

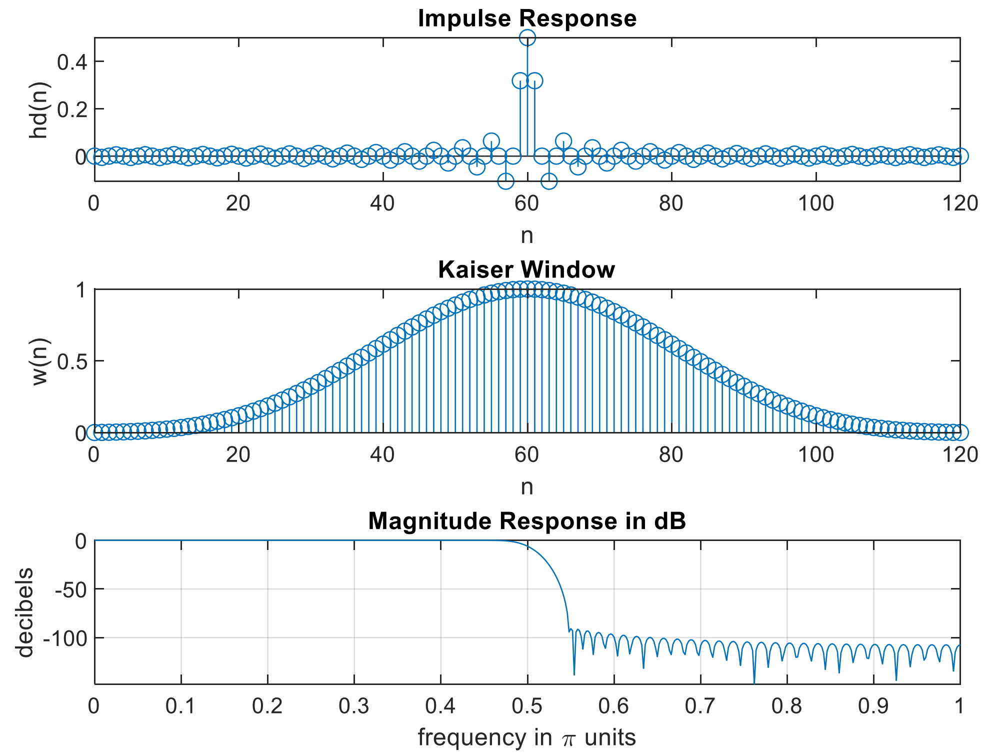
Window Name	Exact Value	Length	Plots
		121	

TABLE 3 Window Method MATLAB Implementation

Window Method	MATLAB Implementation in relevant to table 1 and 2 results
Rectangular (rectw.m)	`% Written and performed by Marwin B. Alejo 2020-20221` `% Return Impulse Response, Rectangular Window, and Magnitude Response plots` `% by simply providing the cut-off frequency in pi*rad and transition width.` `function rectw(wc,tr_width)` `    M=ceil(1.8*pi/tr_width)+1;` `    n=[0:M-1];`

Window Method	MATLAB Implementation in relevant to table 1 and 2 results
	`` alpha=(M-1)/2; m=n-alpha; fc=wc/pi; hd=fc*sinc(fc*m); w_rect=(rectwin(M))'; b=hd.*w_rect; [H,w] = freqz(b,[1],1000,'whole') ; H = (H(1:1:501)); w = (w(1:1:501)); mag = abs(H); db = 20*log10((mag+eps)/max(mag)); %wvtool(b); % for sidelobe measurement figure(); subplot(3,1,1);stem(n,hd);title('Impulse Response');xlabel('n');ylabel('hd(n)'); subplot(3,1,2);stem(n,w_rect);title('Rectangular Window');xlabel('n');ylabel('w(n)'); subplot(3,1,3);plot(w/pi,db);title('Magnitude Response in dB');grid;xlabel('frequency in \pi units');ylabel('decibels'); end ```
Bartlett (bartw.m)	``` % Written and performed by Marwin B. Alejo 2020-20221 % Return Impulse Response, Bartlett Window, and Magnitude Response plots % by simply providing the cut-off frequency in pi*rad and transition width. function bartw(wc,tr_width) M=ceil(6.1*pi/tr_width)+1; n=[0:M-1]; alpha=(M-1)/2; m=n-alpha; fc=wc/pi; hd=fc*sinc(fc*m); w_bart=(bartlett(M))'; b=hd.*w_bart; [H,w] = freqz(b,[1],1000,'whole') ; H = (H(1:1:501)); w = (w(1:1:501)); mag = abs(H); db = 20*log10((mag+eps)/max(mag)); %wvtool(b); % for sidelobe measurement figure(); subplot(3,1,1);stem(n,hd);title('Impulse Response');xlabel('n');ylabel('hd(n)'); subplot(3,1,2);stem(n,w_bart);title('Bartlett Window');xlabel('n');ylabel('w(n)'); subplot(3,1,3);plot(w/pi,db);title('Magnitude Response in dB');grid;xlabel('frequency in \pi units');ylabel('decibels'); ```

Window Method	MATLAB Implementation in relevant to table 1 and 2 results
	`end`
Blackman (blkw.m)	`% Written and performed by Marwin B. Alejo 2020-20221` `% Return Impulse Response, Blackman Window, and Magnitude Response plots` `% by simply providing the cut-off frequency in pi*rad and transition width.` `function blkw(wc,tr_width)` `    M=ceil(11*pi/tr_width)+1;` `    n=[0:M-1];` `    alpha=(M-1)/2;` `    m=n-alpha;` `    fc=wc/pi;` `    hd=fc*sinc(fc*m);` `    w_blk=(blackman(M))';` `    b=hd.*w_blk;` `    [H,w] = freqz(b,[1],1000,'whole') ;` `    H = (H(1:1:501));` `    w = (w(1:1:501));` `    mag = abs(H);` `    db = 20*log10((mag+eps)/max(mag));` `    %wvtool(b); % for sidelobe measurement` `    figure();` `    subplot(3,1,1);stem(n,hd);title('Impulse Response');xlabel('n');ylabel('hd(n)');` `    subplot(3,1,2);stem(n,w_blk);title('Blackman Window');xlabel('n');ylabel('w(n)');` `    subplot(3,1,3);plot(w/pi,db);title('Magnitude Response in dB');grid;xlabel('frequency in \pi` `units');ylabel('decibels');` `end`
Hanning (hannw.m)	`% Written and performed by Marwin B. Alejo 2020-20221` `% Return Impulse Response, Hanning Window, and Magnitude Response plots` `% by simply providing the cut-off frequency in pi*rad and transition width.` `function hannw(wc,tr_width)` `    M=ceil(6.2*pi/tr_width)+1;` `    n=[0:M-1];` `    alpha=(M-1)/2;` `    m=n-alpha;` `    fc=wc/pi;` `    hd=fc*sinc(fc*m);` `    w_hann=(hann(M))';` `    b=hd.*w_hann;` `    [H,w] = freqz(b,[1],1000,'whole') ;` `    H = (H(1:1:501));` `    w = (w(1:1:501));` `    mag = abs(H);`

Window Method	MATLAB Implementation in relevant to table 1 and 2 results
	`db = 20*log10((mag+eps)/max(mag));` `%wvtool(b); % for sidelobe measurement` `figure();` `subplot(3,1,1);stem(n,hd);title('Impulse Response');xlabel('n');ylabel('hd(n)');` `subplot(3,1,2);stem(n,w_hann);title('Hanning Window');xlabel('n');ylabel('w(n)');` `subplot(3,1,3);plot(w/pi,db);title('Magnitude Response in dB');grid;xlabel('frequency in \pi units');ylabel('decibels');` `end`
Hamming (hammw.m)	`% Written and performed by Marwin B. Alejo 2020-20221` `% Return Impulse Response, Hamming Window, and Magnitude Response plots` `% by simply providing the cut-off frequency in pi*rad and transition width.` `function hammw(wc,tr_width)` `M=ceil(6.6*pi/tr_width)+1;` `n=[0:M-1];` `alpha=(M-1)/2;` `m=n-alpha;` `fc=wc/pi;` `hd=fc*sinc(fc*m);` `w_hamm=(hamming(M))';` `b=hd.*w_hamm;` `[H,w] = freqz(b,[1],1000,'whole') ;` `H = (H(1:1:501));` `w = (w(1:1:501));` `mag = abs(H);` `db = 20*log10((mag+eps)/max(mag));` `%wvtool(b); % for sidelobe measurement` `figure();` `subplot(3,1,1);stem(n,hd);title('Impulse Response');xlabel('n');ylabel('hd(n)');` `subplot(3,1,2);stem(n,w_hamm);title('Hamming Window');xlabel('n');ylabel('w(n)');` `subplot(3,1,3);plot(w/pi,db);title('Magnitude Response in dB');grid;xlabel('frequency in \pi units');ylabel('decibels');` `end`
Kaiser (kaisw.m)	`% Written and performed by Marwin B. Alejo 2020-20221` `% Return Impulse Response, Kaiser Window, and Magnitude Response plots` `% by simply providing the cut-off frequency in pi*rad and transition width.` `function kaisw(wc,tr_width, beta)` `M=ceil(1.8*pi/tr_width)+1;` `n=[0:M-1];` `alpha=(M-1)/2;` `m=n-alpha;` `fc=wc/pi;`

Window Method	MATLAB Implementation in relevant to table 1 and 2 results
	`hd=fc*sinc(fc*m);` `w_kais=(kaiser(M, beta))';` `b=hd.*w_kais;` `[H,w] = freqz(b,[1],1000,'whole') ;` `H = (H(1:1:501));` `w = (w(1:1:501));` `mag = abs(H);` `db = 20*log10((mag+eps)/max(mag));` `%wvtool(b); % for sidelobe measurement` `figure();` `subplot(3,1,1);stem(n,hd);title('Impulse Response');xlabel('n');ylabel('hd(n)');` `subplot(3,1,2);stem(n,w_kais);title('Kaiser Window');xlabel('n');ylabel('w(n)');` `subplot(3,1,3);plot(w/pi,db);title('Magnitude Response in dB');grid;xlabel('frequency in \pi units');ylabel('decibels');` `end`

C. Design a digital FIR lowpass filter with specs: $w_p = 0.2\pi, w_s = 0.3\pi, R_p = 0.25dB, A_s = 50dB$

```
function hd=ideal_lp(wc,M)
%hd: ideal LPF impulse response between 0 and M-1
%wc: cut-off frequencies in radians
%M: length of the filter
    alpha=(M-1)/2;
    n=[0: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);
end

C_wp=0.2*pi; C_ws=0.3*pi; C_tr_width=C_ws-C_wp;
C_M=ceil(6.6*pi/C_tr_width)+1;
C_n=[0:C_M-1];
C_wc=(C_ws+C_wp)/2; %ideal cutoff frequency
C_hd=ideal_lp(C_wc,C_M);
C_w_hamming=(hamming(C_M))';
C_h=C_hd.*C_w_hamming;
[C_db,C_mag,C_pha,C_grd,C_w]=freqz_m(C_h,[1]);
figure();
subplot(3,1,1);stem(C_n,C_h); title('Impulse Response');xlabel('n');ylabel('hd(n)');
subplot(3,1,2);stem(C_n,C_w_hamming);title('Hamming Window');xlabel('n');ylabel('w(n)');
subplot(3,1,3);plot(C_w/pi,C_db);title('Magnitude Response in dB');grid;ylabel('decibel');
xlabel('frequency in \pi units');
```

For this section to be done, two different functions were written: ideal_lp and freqz_m. The ideal_lp function returns the coefficients of an ideal LPF of order n and length M. The freqz_m function is a modified freqz function and return the relative magnitude, and absolute magnitude values of an ideal LPF.

Unlike in section B, cut-off frequency is not provided hence, among of the hidden tasks is to determine the unknown parameters to design a lowpass filter based on specification (refer to activity 6 manual for specs). This also include the length M, order number n, and transition width which are reflected on the first three lines of the code above.

After determining the length M and the ideal cut-off frequency, the next thing to do is determine the ideal LPF coefficients (hd) by using the ideal_lp function. Provided that the design sideband attenuation is 50dB, we will be using Hamming window method as its peak sideband attenuation is closer to the desired 50dB specification (refer to table 1 above) rather than when using either Kaiser or Hann methods. Using the hamming method requires that the length M be determined first. The value of M is measured using the same formula used in hammw.m and in table 1.

Since 0.25dB ripple attenuation is not necessary in the development of a filter in this section, it is used to determine whether the generated filter is correct or not.

The impulse response, hamming, and magnitude response in dB values are generated using the freqz_m function. Figure 1 below shows the plot of the above MATLAB implementation. It is also shown that the desired LPF specifications are met with the value of ripple attenuation is equal to the peak LPF coefficient in impulse response plot.

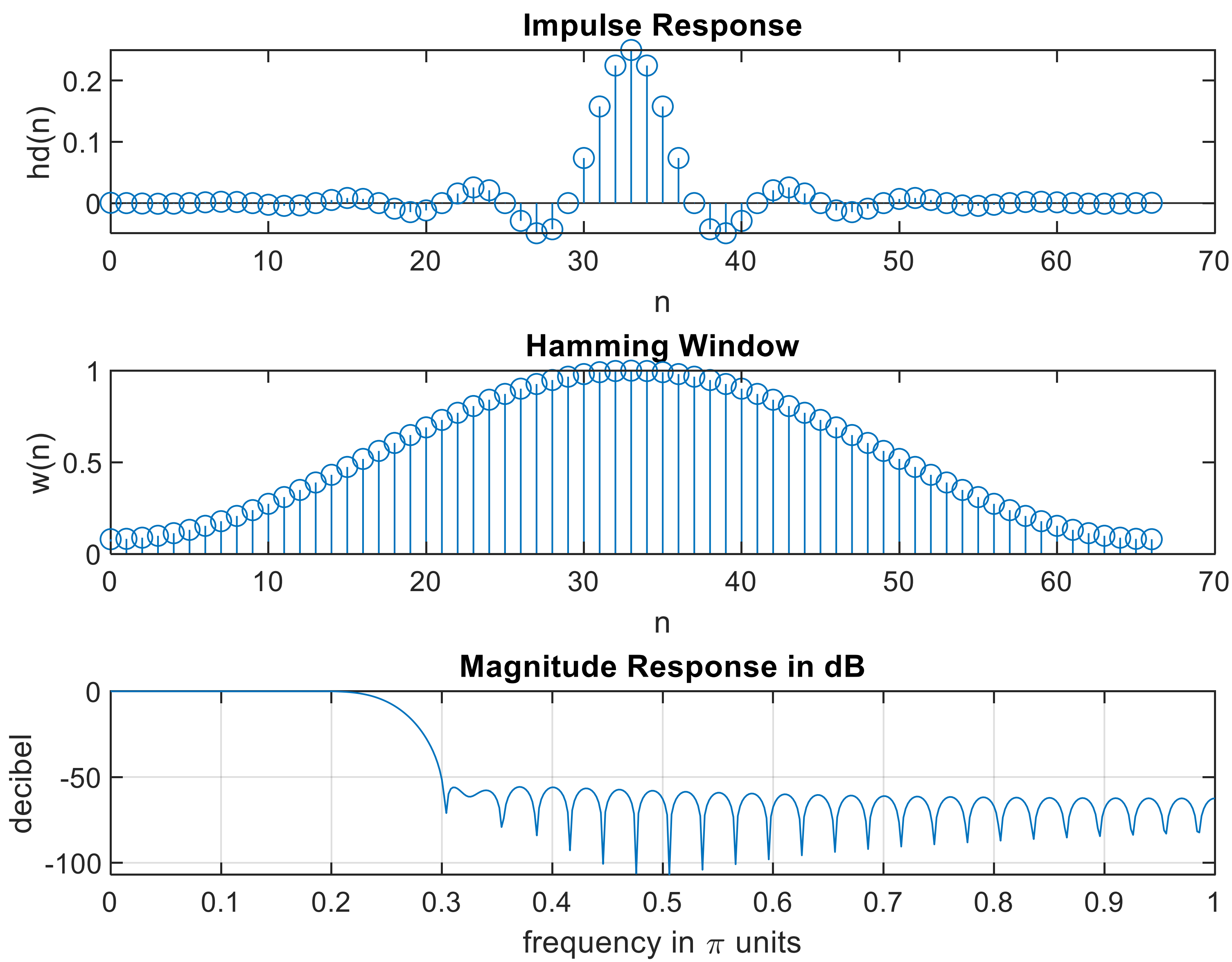


Figure 1 Impulse Response (Top), Hamming Window (Middle), Magnitude Response (Bottom) Plots

D. Frequency Sampling (for grads only)

$$H(k) = \begin{cases} (-1)^k & 0 \le k \le 7 \\ 0.5 & k = 8 \\ 0 & 9 \le k \le 23 \\ 0.5 & k = 24 \\ (-1)^k & 25 \le k \le 31 \end{cases}$$

```
D_f = [0 0.125 0.1875 0.25 0.3125 0.375 0.4375 0.5 0.5 0.5625 0.625 0.6875 0.75 0.8125 0.875 0.9375 1];
D_m = [1 -1 1 -1 1 -1 1 -1 0.5 0 0 0 0 0 0 0 0];
D_b1 = fir2(32,D_f,D_m);

D_f = [0 0.5 0.5 1];
D_m = [1 0.5 0 0];
D_b1 = fir2(32,D_f,D_m);
freqz(D_b1,1,[]);
```

Shown in figure 2 is the frequency response using frequency sampling as described by the samples in H(k) above.

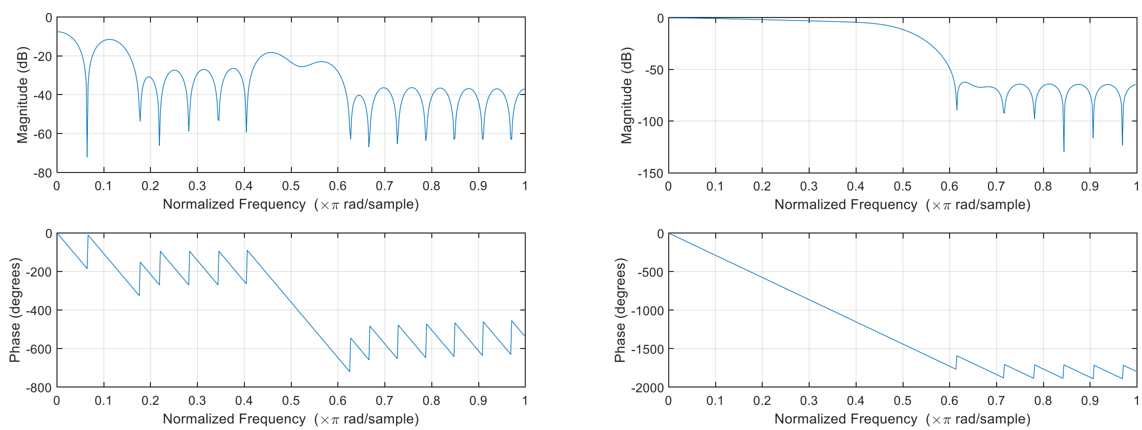


Figure 2 Frequency Response of H(k) using Frequency Sampling method.

a. What is the order of the FIR filter? Is the filter linear phase?

The FIR filter is of 32 order. Regardless of phase discontinuities which indicates that the filter has sign reversal components, it is linear phase.

b. Compare the transition bandwidth, passband, and stopband ripple heights of FIR filter design using windowing method and frequency sampling methods of the same filter order.

The frequency response of H(k) is like the frequency response plot a hamming-based FIR filter of the same length except that the passband is shorter in frequency sampling than in windowing method, the transition band is wider is frequency sampling than in windowing method, and stopband ripple heights are more optimal in windowing method than in frequency sampling method. This is in reflection of the fact below that frequency sampling method show the exact solution but is ineffective on selective precision points.

c. What is the main advantage of using frequency sampling methods with windowing methods?

Frequency sampling method is effective and practical on applications that require an exact solution at discrete frequency locations in comparison with windowing method. Frequency sampling is very simple yet impractical except on selective precision points moreover, it will be exact on frequency sample location with ripple in between.