National Institute of Technology
Calicut

Department of Electronics and Communication
Engineering

Digital Signal Processing LAB

Week 6 Report

Date: 25-09–2025
Instructor: Dr. Sakthivel V

Submitted by:

Mummana Jagadeesh B231113EC
Muhammed Nihal MP B230437EC

Aim

To implement FIR filters of different types including lowpass, highpass, bandpass, and bandstop by applying window functions like Rectangular, Bartlett, Hanning, Hamming, and Blackman, and to study their impulse and frequency domain behavior in MATLAB.

Theory

Finite Impulse Response (FIR) filters are frequently used in digital signal processing due to their inherent stability and ability to achieve linear phase. The frequency response of an FIR filter is determined by its impulse response $h[n]$.

In the case of ideal filters such as Low Pass, High Pass, Band Pass, and Band Stop, the impulse responses are infinitely long, which makes them non-realizable. To implement these filters, the windowing method is applied. In this method, the ideal response $h_d[n]$ is multiplied with a finite-duration window $w[n]$, resulting in a realizable FIR filter:

$$h[n] = h_d[n] \times w[n]$$

Each type of window influences the filter characteristics in a different way:

- **Rectangular Window:** Basic window with sharp edges.
- **Modified Rectangular Window:** Slight adjustment to reduce error.
- **Hanning Window:** Smooth tapering at the ends.
- **Hamming Window:** Similar to Hanning, but with better side lobe control.
- **Bartlett Window:** Triangular shape, linearly decreasing.
- **Blackman Window:** Wider main lobe, strong side lobe reduction.

Process of FIR Filter Design

The general procedure for designing an FIR filter can be summarized in the following steps:

1. Define the filter specifications such as passband frequency (ω_p), stopband frequency (ω_s), passband ripple (r_p), and stopband attenuation (r_s).
2. Estimate the filter order using an appropriate empirical relation.
3. Derive the ideal impulse response $h_d[n]$ corresponding to the desired filter type (LPF, HPF, BPF, or BSF).
4. Choose a suitable window function $w[n]$.
5. Obtain the practical filter response by multiplying:

$$h[n] = h_d[n] \cdot w[n]$$

6. Analyze the filter by plotting its magnitude and phase response.

Window Method for FIR Filter Design

Apart from the Window Method, other commonly used FIR design approaches include the Frequency Sampling Method and the Optimal (Equiripple) Method. In this report, we focus on the Window Method due to its simplicity and effectiveness for many practical applications.

In the Window Method, the infinite-length ideal impulse response $h_d[n]$ is truncated using a finite-duration window function $w[n]$ to obtain a realizable FIR filter:

$$h[n] = h_d[n] \cdot w[n]$$

Different window functions provide different trade-offs between main lobe width and side lobe attenuation. The commonly used windows are described below.

- **Rectangular Window:** The simplest window, equal weighting for all samples.

$$w[n] = 1, \quad 0 \leq n \leq N-1$$

- **Modified Rectangular Window:** A slight modification of the rectangular window to reduce ripple by adjusting the first and last coefficients.

$$w[n] = \begin{cases} 0.5, & n = 0 \text{ or } n = N-1 \\ 1, & 1 \leq n \leq N-2 \end{cases}$$

- **Hanning Window:** Provides smooth tapering at the ends to reduce side lobes.

$$w[n] = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1$$

- **Hamming Window:** Similar to Hanning but with slightly better side lobe suppression. The general form uses a parameter α, commonly taken as 0.54.

$$w[n] = \alpha - (1-\alpha)\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1$$

- **Bartlett Window:** Triangular shape, linearly decreasing to zero at the ends.

$$w[n] = \begin{cases} \frac{2n}{N-1}, & 0 \leq n \leq \frac{N-1}{2} \\ 2 - \frac{2n}{N-1}, & \frac{N-1}{2} < n \leq N-1 \end{cases}$$

- **Blackman Window:** Smooth window with strong side lobe suppression and a wider main lobe.

$$w[n] = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right), \quad 0 \leq n \leq N-1$$

Mathematical Expressions of Ideal FIR Filter Responses

The ideal impulse response $h_d[n]$ represents the theoretical, infinite-length response of FIR filters. These responses are truncated using window functions to obtain practical FIR filters. The impulse responses for common filter types are given below. Let $\alpha = \frac{N-1}{2}$, where N is the filter length.

- **Low-Pass Filter (LPF):** Passes frequencies below the cutoff frequency ω_p.

$$h_d[n] = \begin{cases} \frac{\sin(\omega_p(n-\alpha))}{\pi(n-\alpha)}, & n \neq \alpha \\ \frac{\omega_p}{\pi}, & n = \alpha \end{cases}$$

- **High-Pass Filter (HPF):** Attenuates frequencies below the cutoff ω_p.

$$h_d[n] = \begin{cases} -\frac{\sin(\omega_p(n-\alpha))}{\pi(n-\alpha)}, & n \neq \alpha \\ 1 - \frac{\omega_p}{\pi}, & n = \alpha \end{cases}$$

- **Band-Pass Filter (BPF):** Passes frequencies between ω_{p1} and ω_{p2}.

$$h_d[n] = \begin{cases} \frac{\sin(\omega_{p2}(n-\alpha))-\sin(\omega_{p1}(n-\alpha))}{\pi(n-\alpha)}, & n \neq \alpha \\ \frac{\omega_{p2}-\omega_{p1}}{\pi}, & n = \alpha \end{cases}$$

- **Band-Stop Filter (BSF):** Attenuates frequencies between ω_{p1} and ω_{p2}.

$$h_d[n] = \begin{cases} \frac{\sin(\omega_{p1}(n-\alpha))-\sin(\omega_{p2}(n-\alpha))}{\pi(n-\alpha)}, & n \neq \alpha \\ 1 - \frac{\omega_{p2}-\omega_{p1}}{\pi}, & n = \alpha \end{cases}$$

FIR Filter Order Estimation

The order of an FIR filter determines the number of coefficients N and directly affects its frequency response. A higher order generally provides a sharper transition between passband and stopband. The filter order can be estimated using empirical formulas based on the filter specifications.

Let the transition width be

$$\Delta\omega = |\omega_s - \omega_p|$$

and let r_p and r_s denote the passband ripple and stopband attenuation, respectively. An empirical formula for estimating the minimum filter order N is:

$$N \approx \frac{-10\log_{10}(r_p r_s) - 15}{14\Delta\omega}$$

This formula provides a reasonable starting point for practical FIR filter design. After estimating N, it is often rounded up to the nearest integer to ensure the design meets the specifications.

MATLAB Code

MATLAB CODE

```
clc; clear; close all;

%% === User Inputs ===
fprintf('--- FIR Filter Design using Window Method ---\n');
pass_ripple = input('Enter passband ripple: ');
stop_att = input('Enter stopband attenuation: ');
wp = input('Enter passband frequency: ');
ws = input('Enter stopband frequency: ');

%% === Estimate Filter Order ===
delta_w = abs(ws - wp);
if delta_w <= 0
    error('Stopband frequency must be greater than passband frequency.');
end

N_est = ceil((-10*log10(pass_ripple*stop_att) - 15)/(14*delta_w));
if mod(N_est,2)==0
    N_est = N_est + 1;
end
fprintf('Estimated Filter Order: N = %d\n', N_est);

tau = (N_est-1)/2;
n = 0:N_est-1;

%% === Filter and Window Types ===
filt_types = {'lowpass','highpass','bandpass','bandstop'};
win_types = {'rectangular','modified_rect','bartlett','hanning','hamming','blackman'};

colors = lines(length(win_types));

for f = 1:length(filt_types)

    num_windows = length(win_types);
    hd_cell = cell(1,num_windows);
    freq_cell = cell(1,num_windows);
    win_cell = cell(1,num_windows);

    % Precompute impulse & frequency responses
    for w_idx = 1:num_windows
```

```
    % —— Ideal impulse response ——
    switch filt_types{f}
        case 'lowpass'
            hd = hd_lpf(n, tau, wp);
        case 'highpass'
            hd = hd_hpf(n, tau, wp);
        case 'bandpass'
            if wp >= ws, error('Lower cutoff wp must be
                < upper cutoff ws.'); end
            hd = hd_bpf(n, tau, wp, ws);
        case 'bandstop'
            if wp >= ws, error('Lower cutoff wp must be
                < upper cutoff ws.'); end
            hd = hd_bsf(n, tau, wp, ws);
    end

    % —— Window selection ——
    switch win_types{w_idx}
        case 'rectangular', win = rect_win(n, N_est);
        case 'modified_rect', win = mod_rect_win(n,
           N_est);
        case 'bartlett', win = bartlett_win(n, N_est);
        case 'hanning', win = hanning_win(n, N_est);
        case 'hamming', win = hamming_win(n, N_est);
        case 'blackman', win = blackman_win(n, N_est);
    end

    h_fir = hd .* win;

    % —— Frequency response (manual FFT) ——
    H_f = manual_fft(h_fir, 2048); % custom FFT
    freq_axis = linspace(-pi, pi, length(H_f))/pi * 2;
       % from -2 to 2

    freq_cell{w_idx} = H_f;
    hd_cell{w_idx} = h_fir;
    win_cell{w_idx} = win;
end

%% —— Plot in 6x3 layout (Phase, Magnitude, Window
   function) ——
figure('Name', [filt_types{f} ' Filter Responses'], '
   NumberTitle','off');
sgtitle([filt_types{f} ' Filter Responses']);

for w_idx = 1:num_windows
```

```
        % subplot indices
        phase_idx = (w_idx-1)*3 + 1;
        mag_idx   = (w_idx-1)*3 + 2;
        win_idx   = (w_idx-1)*3 + 3;

        freq_axis = linspace(-pi, pi, length(freq_cell{
           w_idx}))/pi * 2; % -2 to 2

        % === Phase response (manual) ===
        subplot(6,3,phase_idx);
        plot(freq_axis, unwrap_manual_phase(freq_cell{w_idx
           }), ...
             'Color', colors(w_idx,:), 'LineWidth', 1.5);
        title([win_types{w_idx} ' Phase'],'Interpreter','
           none');
        xlabel('\omega / \pi'); ylabel('Phase (rad)'); grid
             on;
        xlim([-2 2]);

        % === Magnitude response (manual) ===
        subplot(6,3,mag_idx);
        plot(freq_axis, manual_abs(freq_cell{w_idx}), ...
             'Color',colors(w_idx,:),'LineWidth',2);
        title(sprintf('%s |H(\\omega)|', win_types{w_idx}),
             'Interpreter','none');
        xlabel('\omega / \pi'); ylabel('Magnitude'); grid
           on;
        ylim([0 1.2*max(manual_abs(freq_cell{w_idx}))]);
        xlim([-2 2]);

        % === Window function ===
        subplot(6,3,win_idx);
        stem(n, win_cell{w_idx}, 'filled', 'Color', colors(
           w_idx,:));
        title([win_types{w_idx} ' Window'],'Interpreter','
           none');
        xlabel('n'); ylabel('w[n]'); grid on;
    end
end

%% ====== Ideal Impulse Response Functions ======
function hd = hd_lpf(n, tau, wp)
    hd = zeros(size(n));
    for i=1:length(n)
        if n(i)~=tau, hd(i)=sin(wp*(n(i)-tau))/(pi*(n(i)-
           tau));
```

```
        else, hd(i)=wp/pi; end
    end
end

function hd = hd_hpf(n, tau, wp)
    hd = zeros(size(n));
    for i=1:length(n)
        if n(i)~=tau, hd(i)=-sin(wp*(n(i)-tau))/(pi*(n(i)-
           tau));
        else, hd(i)=1-wp/pi; end
    end
end

function hd = hd_bpf(n, tau, wp1, wp2)
    hd = zeros(size(n));
    for i=1:length(n)
        if n(i)~=tau, hd(i)=(sin(wp2*(n(i)-tau))-sin(wp1*(n
           (i)-tau)))/(pi*(n(i)-tau));
        else, hd(i)=(wp2-wp1)/pi; end
    end
end

function hd = hd_bsf(n, tau, wp1, wp2)
    hd = zeros(size(n));
    for i=1:length(n)
        if n(i)~=tau, hd(i)=(sin(wp1*(n(i)-tau))-sin(wp2*(n
           (i)-tau)))/(pi*(n(i)-tau));
        else, hd(i)=1-(wp2-wp1)/pi; end
    end
end

%% ======= Window Functions =======
function w = rect_win(n,N), w = ones(size(n)); end
function w = mod_rect_win(n,N), w = ones(size(n)); w(1)
   =0.5; w(end)=0.5; end
function w = bartlett_win(n,N)
    w = zeros(size(n));
    for i=1:length(n)
        if n(i)<=(N-1)/2, w(i)=2*n(i)/(N-1); else, w(i)
           =2-2*n(i)/(N-1); end
    end
end
function w = hanning_win(n,N), w = 0.5*(1-cos(2*pi*n/(N-1))
   ); end
function w = hamming_win(n,N)
    alpha = 0.54; w = alpha-(1-alpha)*cos(2*pi*n/(N-1));
```

```
end
function w = blackman_win(n,N)
    w = 0.42-0.5*cos(2*pi*n/(N-1))+0.08*cos(4*pi*n/(N-1));
end

%% ======== Manual FFT ========
function X = manual_fft(x, N)
    % zero-pad if needed
    x = [x zeros(1,N-length(x))];
    X = zeros(1,N);
    for k = 0:N-1
        sum_val = 0;
        for n = 0:N-1
            sum_val = sum_val + x(n+1)*exp(-1j*2*pi*k*n/N);
        end
        X(k+1) = sum_val;
    end
    % Shift zero-frequency component to center
    X = fftshift_manual(X);
end

function Xs = fftshift_manual(X)
    N = length(X);
    p = floor(N/2);
    Xs = [X(p+1:end) X(1:p)];
end

%% ======== Manual Magnitude and Phase ========
function mag = manual_abs(X)
    mag = sqrt(real(X).^2 + imag(X).^2);
end

function ph = unwrap_manual_phase(X)
    ph = angle_manual(X);
    for i=2:length(ph)
        while ph(i)-ph(i-1) > pi, ph(i:end) = ph(i:end) -
           2*pi; end
        while ph(i)-ph(i-1) < -pi, ph(i:end) = ph(i:end) +
           2*pi; end
    end
end

function ph = angle_manual(X)
    ph = atan2(imag(X), real(X));
end
```

Code Implementation

The MATLAB program implements FIR filter design using the windowing method. The steps followed in the code are summarized below:

1. **User Input:** The program prompts the user to input the passband ripple, stopband attenuation, passband frequency (ω_p), and stopband frequency (ω_s). These values define the filter specifications.

2. **Filter Order Estimation:** The transition bandwidth is calculated as $\Delta\omega = |\omega_s - \omega_p|$. The filter order N is estimated using an empirical formula:

$$N = \left\lceil \frac{-10\log_{10}(\delta_p \cdot \delta_s) - 15}{14\Delta\omega} \right\rceil$$

   where δ_p and δ_s represent the passband ripple and stopband attenuation respectively. If N is even, it is incremented by 1 to maintain a symmetric impulse response.

3. **Time Vector and Center Index:** The discrete-time vector $n = 0 : N - 1$ is generated, and the center index $\tau = (N-1)/2$ is computed for designing symmetric FIR filters.

4. **Filter and Window Selection:** The code considers four types of filters: lowpass, highpass, bandpass, and bandstop. Six window functions are available: Rectangular, Modified Rectangular, Bartlett, Hanning, Hamming, and Blackman. For each combination of filter type and window, the following steps are performed:

   (a) **Ideal Impulse Response:** The ideal impulse response $h_d[n]$ for the selected filter type is computed. The formulas handle the special case $n = \tau$ separately to avoid division by zero.

   (b) **Window Application:** The selected window function $w[n]$ is multiplied element-wise with the ideal impulse response to obtain the practical FIR filter:

$$h[n] = h_d[n] \cdot w[n]$$

   Each window type has a dedicated function, including a piecewise definition for the Bartlett window and a parameterized α for the Hamming window.

   (c) **Frequency Response:** The frequency response $H(\omega)$ is computed using the Fast Fourier Transform (FFT) of the windowed impulse response. Only the first half of the FFT is used since the response is symmetric.

5. **Storage for Plotting:** The impulse and frequency responses are stored in separate arrays for later visualization.

6. **Visualization:** Finally, the computed impulse responses and frequency responses are plotted for each window and filter type to allow analysis of the FIR filter characteristics. Each plot demonstrates how the choice of window affects the filter behavior.

This implementation allows flexible experimentation with different FIR filter types and windows, providing insights into the effect of windowing on filter performance.

Output

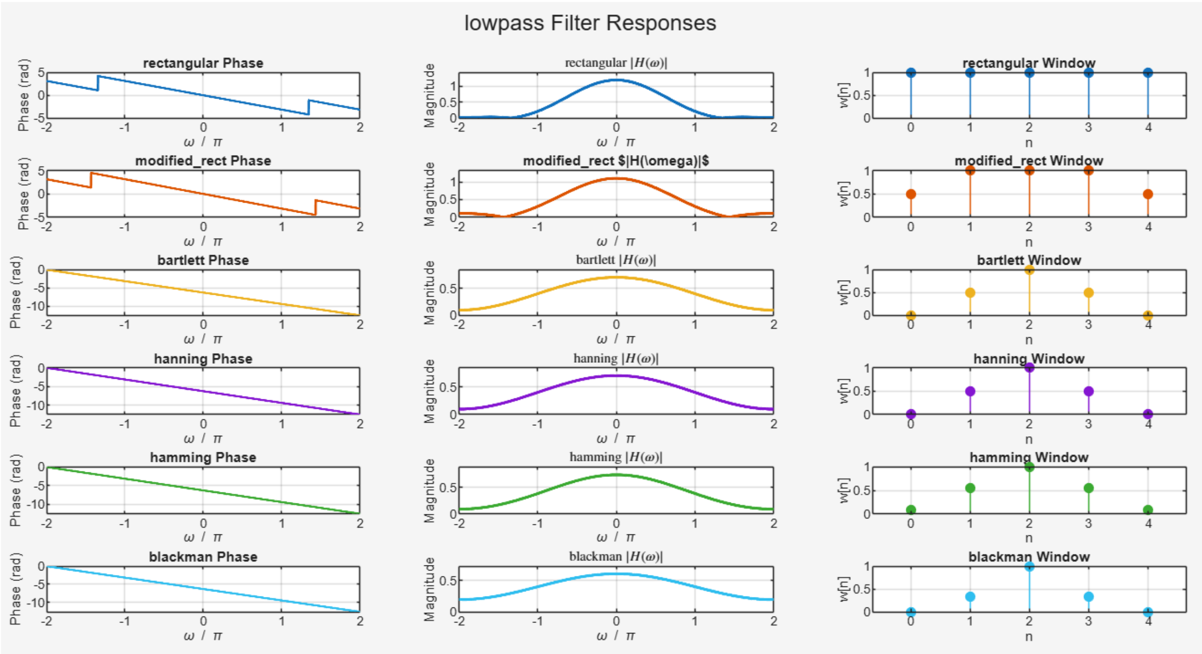

Figure 1: Low Pass Filter Responses - MATLAB Plot

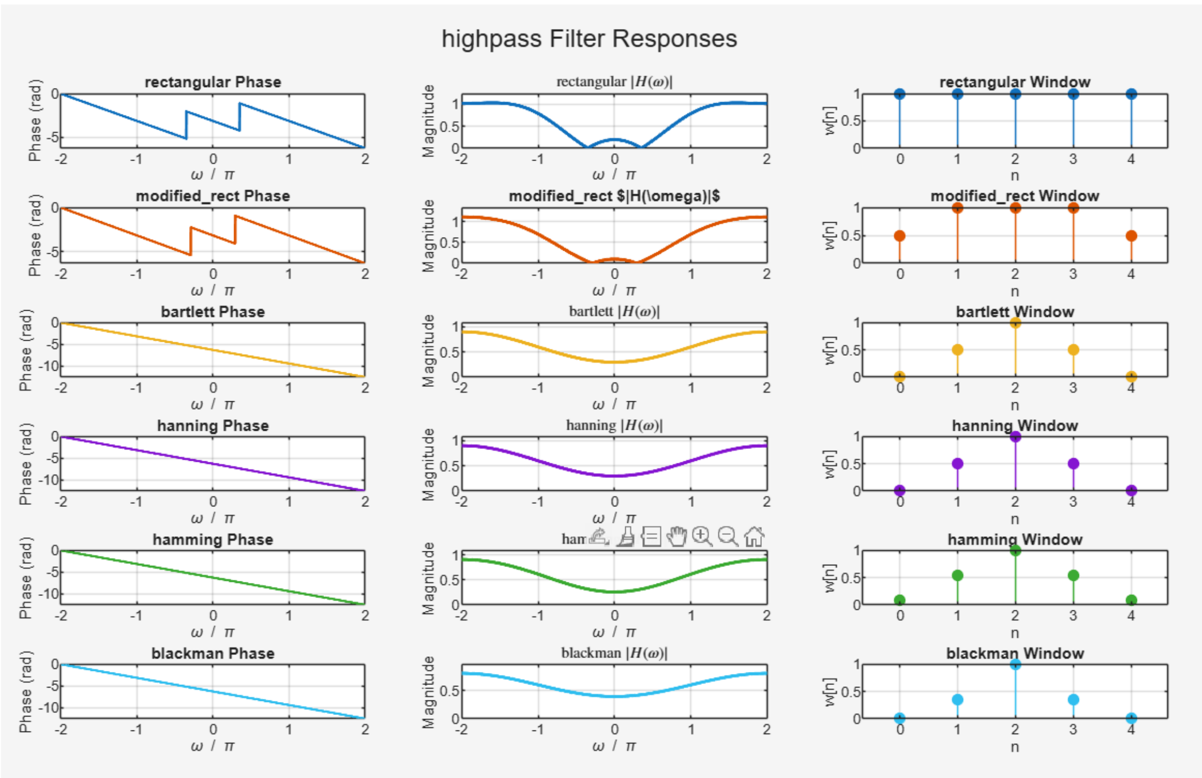

Figure 2: High Pass Filter Responses - MATLAB Plot

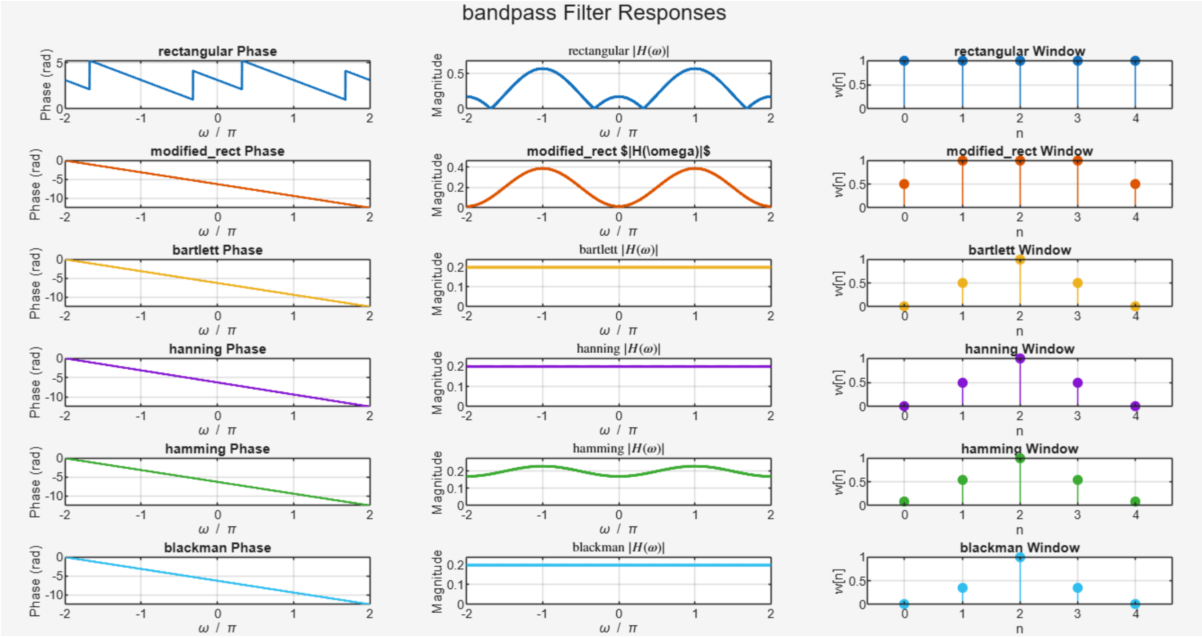

Figure 3: Band Pass Filter Responses - MATLAB Plot

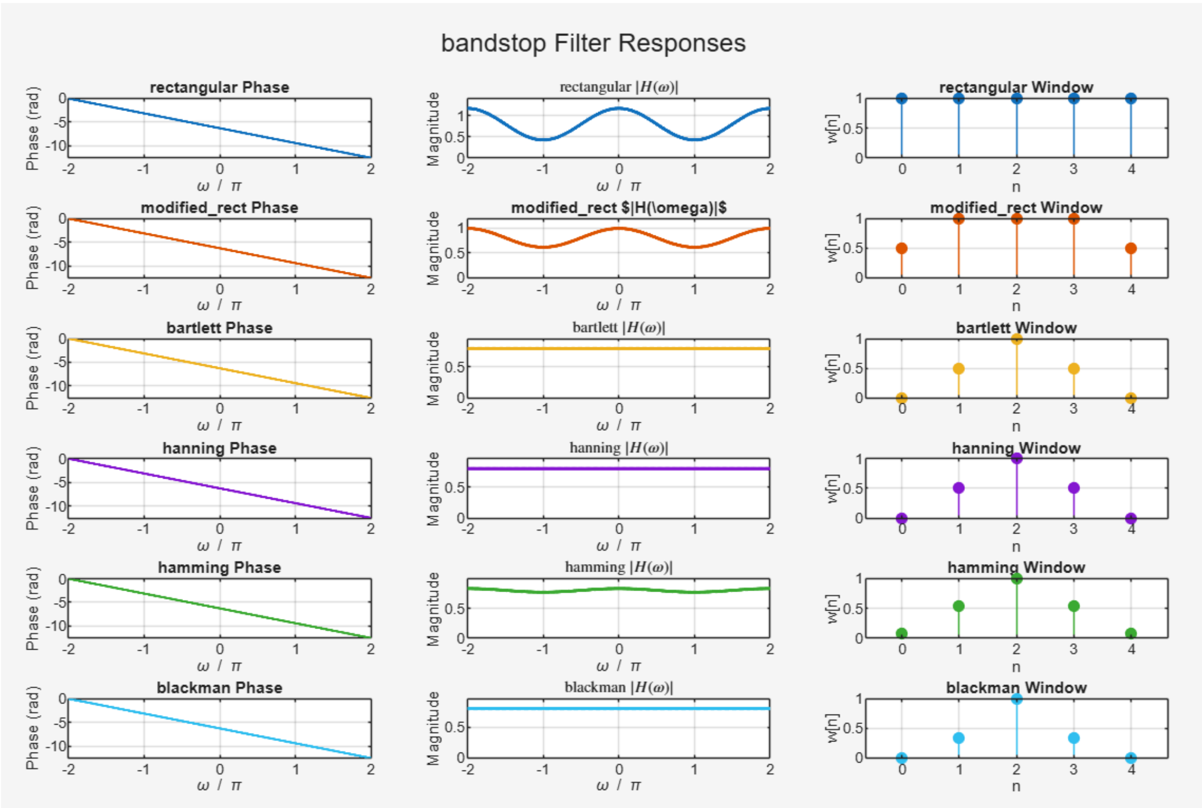

Figure 4: Band Stop Filter Responses - MATLAB Plot

```
--- FIR Filter Design using Window Method
Enter passband ripple: 0.01
Enter stopband attenuation: 0.001
Enter passband frequency: 0.4*pi
Enter stopband frequency: 0.6*pi
Estimated Filter Order: N = 9
>>
```

Figure 5: Command Window Output

Observations

- Rectangular window produces a narrow main lobe but exhibits high side lobes, leading to poor stopband attenuation.
- Bartlett and Hanning windows reduce side lobes at the cost of slightly wider main lobes, providing moderate attenuation improvement.
- Hamming and Blackman windows offer better stopband suppression with slightly wider transition bands compared to Rectangular and Bartlett windows.
- Low-pass filters: Rectangular window shows sharp transitions but poor stopband attenuation, while Blackman provides deeper suppression with smoother transitions.
- High-pass filters: Similar trade-offs are observed; Hamming and Blackman achieve improved attenuation.
- Band-pass filters: Narrow passbands are handled more effectively by Hamming and Blackman, whereas Rectangular may result in leakage.
- Band-stop filters: Blackman window achieves the deepest stopband attenuation; Hamming provides a balance between transition width and attenuation.

Conclusion

- FIR filters can be effectively designed using the windowing method for low-pass, high-pass, band-pass, and band-stop filters.
- The choice of window significantly affects main lobe width, side lobe levels, transition band sharpness, and stopband attenuation.
- Rectangular windows offer sharp transitions but poor suppression of unwanted frequencies, while Hamming and Blackman windows improve stopband attenuation at the expense of slightly wider transition bands.
- MATLAB simulations confirm theoretical expectations and demonstrate the practical impact of different window functions on FIR filter performance.