SDM COLLEGE OF ENGINEERING AND TECHNOLOGY

DHARWAD - 580002

Department of Electrical and Electronics Engineering

CTA ASSIGNMENT

AcademicYear:2024-25

Course: Digital Signal Processing

CourseInstructor: Prof.ManjulaS.Sureban

Name:SHRADDHA G MANE

USN:2SD22EE041

IIR Butterworth BP Filter

Problem statement :

Q)Design and generate IIR Butterworth BP Filter using MATLAB

Solution:-

The goal of the task "Design and generate IIR Butterworth BP Filter using MATLAB" is to create a band-pass filter with specific characteristics using MATLAB. The Butterworth filter is chosen for its maximally flat frequency response in the passband, ensuring minimal distortion. The design involves defining the filter specifications such as passband frequency range, sampling frequency, and filter order. Using MATLAB, the filter coefficients are calculated, and the filter is implemented to process signals, allowing frequencies within the passband to pass through while attenuating others. This exercise helps understand digital filter design principles and their practical applications in signal processing.

Code:-

```
% Specifications
wsl = input("enter lower stopband edge ");  % Lower stopband edge
wpl = input("enter lower passband edge ");  % Lower passband edge
wpu =  input("enter upper passband edge ");  % Upper passband edge
wsu = input("enter upper stopband edge ");  % Upper stopband edge
Ap = input("enter passband attenuation ");        % Passband attenuation in dB
As = input("enter stopband attenuation ");        % Stopband attenuation in dB

% Step 1: Pre-warp digital frequencies using Bilinear Transform
T = 1;  % Sampling period (normalized to 1)
Omega_sl = 2/T * tan(wsl / 2);
Omega_pl = 2/T * tan(wpl / 2);
Omega_pu = 2/T * tan(wpu / 2);
Omega_su = 2/T * tan(wsu / 2);

% Step 2: Compute center frequency and bandwidth for bandpass
Omega_0 = sqrt(Omega_pl * Omega_pu);  % Center frequency
B = Omega_pu - Omega_pl;              % Bandwidth

% Step 3: Map bandpass filter to lowpass prototype
Omega_Ls1 = (Omega_sl^2 - Omega_0^2) / (B * Omega_sl);  % Lower stopband
Omega_Ls2 = (Omega_su^2 - Omega_0^2) / (B * Omega_su);  % Upper stopband
Omega_Ls = min(abs(Omega_Ls1), abs(Omega_Ls2));         % Worst-case stopband

% Step 4: Calculate filter order N
epsilon = sqrt(10^(Ap/10) - 1);
A = sqrt(10^(As/10) - 1);
N = ceil(log10(A / epsilon) / (2 * log10(Omega_Ls)));  % Filter order
disp(['Filter Order N = ', num2str(N)]);

% Step 5: Find lowpass Butterworth poles
poles = zeros(1, N);
for k = 1:N
    theta = (2*k - 1) * pi / (2*N);  % Angle for pole location
    poles(k) = -sin(theta) + 1i * cos(theta);  % Poles in s-plane
```

```
end
disp('Lowpass Prototype Poles:');
disp(poles);

% Step 6: Transform lowpass poles to bandpass poles
bandpass_poles = zeros(1, 2*N);
for k = 1:N
    s = poles(k);  % Lowpass pole
    % Bandpass transformation: s -> (s/B) + (Omega_0^2 / s)
    bandpass_poles(2*k - 1) = (s*B/2) + sqrt((s*B/2)^2 - Omega_0^2);
    bandpass_poles(2*k) = (s*B/2) - sqrt((s*B/2)^2 - Omega_0^2);
end
disp('Bandpass Poles:');
disp(bandpass_poles);

% Step 7: Apply Bilinear Transform  to map analog poles to digital poles
digital_poles = (2 + bandpass_poles) ./ (2 - bandpass_poles);
disp('Digital Poles:');
disp(digital_poles);

% Step 8: Manually construct transfer function from poles
% Multiply (z - pole) terms to get numerator and denominator
numerator = 1;  % (Assume system gain = 1)
denominator = poly(digital_poles);  % Denominator polynomial from poles

disp('Digital Filter Coefficients:');
disp('Numerator Coefficients:');
disp(numerator);
disp('Denominator Coefficients:');
disp(denominator);

% Step 9: Frequency response of the filter
[H, w] = freqz(numerator, denominator, 1024);
figure;
plot(w/pi, 20*log10(abs(H)));
grid on;
title('Magnitude Response of IIR Butterworth Bandpass Filter');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude (dB)');
```

input values:-

Q)Design A Band Pass Filter With Following Specifications

1. Lower Stop Band Edge, $\Omega sl = 0.1\pi$
2. Lower Pass Band Edge, $\Omega pl = 0.4\pi$
3. Upper Pass Band Edge, $\Omega pu = 0.6\pi$
4. Upper Stop Band Edge, $\Omega su = 0.9\pi$
5. Pass Band Attenuation, Ap = -3 Db
6. Stop Band Attenuation, As = -18 Db

output:-

```
enter lower stopband edge 0.1*pi
enter lower passband edge 0.4*pi
enter upper passband edge  0.6*pi
```

```
enter upper stopband edge  0.9*pi
enter passband attenuation -3
enter stopband attenuation -18
Filter Order N = 1
Lowpass Prototype Poles:
  -1.0000 + 0.0000i

Bandpass Poles:
  -0.6498 - 1.8915i  -0.6498 + 1.8915i

Digital Poles:
   0.0000 - 0.7138i   0.0000 + 0.7138i

Digital Filter Coefficients:
Numerator Coefficients:
     1

Denominator Coefficients:
    1.0000   -0.0000    0.5095
```