**EE 387 – Signal Processing**

**Lab 3: System Functions and Frequency Response**

**E/20/420 – WANASINGHE J.K.**

## Part 1: Pole-Zero Diagrams in MATLAB.

A pole-zero diagram displays the “poles” and “zeros” of the system function by placing an ‘x’ at each pole location and an ‘o’ at each zero location in the complex s-plane. Poles and zeros can be found out by using roots function in MATLAB.

Example: Find out the zeros and poles of the following system function and plot them.

clear all;

close all;

b = [1 -1]; % Numerator coefficients

a = [1 2 2]; % Demoninator coefficients

zs = roots(b); % Generetes Zeros

ps = roots(a); % Generetes poles

% Create pole-zero plot with custom markers and colors

figure;

hold on;

% Plot zeros as white circles

plot(real(zs), imag(zs), 'o', 'MarkerSize', 8, 'MarkerEdgeColor', 'white', 'MarkerFaceColor', 'none', 'LineWidth', 2);

% Plot poles as white crosses

plot(real(ps), imag(ps), 'x', 'MarkerSize', 10, 'Color', 'white', 'LineWidth', 3);

% Customize the plot

grid on;

axis equal;

xlabel('Real Part');

ylabel('Imaginary Part');

title('Pole-Zero Plot');

legend('Zeros (O)', 'Poles (X)', 'Location', 'best');

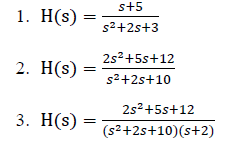
hold off;



Figure 1: Pole-Zero plot of the Example

## Exercise

Using the method given above, find out the zeros and poles of the following system functions and plot them:



b1 = [1 5]; % Numerator coefficients

a1 = [1 2 3]; % Demoninator coefficients

b2 = [2 5 12]; % Numerator coefficients

a2 = [1 2 10]; % Demoninator coefficients

b3 = [2 5 12]; % Numerator coefficients

a3 = conv([1 2 10],[1 2]); % Demoninator coefficients

% Calculate zeros and poles for each transfer function

zs1 = roots(b1);

ps1 = roots(a1);

zs2 = roots(b2);

ps2 = roots(a2);

zs3 = roots(b3);

ps3 = roots(a3);

% Create figure with subplots

figure;

% Subplot 1

subplot(1,3,1);

hold on;

% Plot zeros as white circles

plot(real(zs1), imag(zs1), 'o', 'MarkerSize', 8, 'MarkerEdgeColor', 'white', 'MarkerFaceColor', 'none', 'LineWidth', 2);

% Plot poles as white crosses

plot(real(ps1), imag(ps1), 'x', 'MarkerSize', 10, 'Color', 'white', 'LineWidth', 3);

grid on;

axis equal;

xlim([-6 2]);

ylim([-3 3]);

xlabel('Real Part');

ylabel('Imaginary Part');

title('H1(s): Poles and Zeros');

legend('Zeros (O)', 'Poles (X)', 'Location', 'best');

hold off;

% Subplot 2

subplot(1,3,2);

hold on;

% Plot zeros as white circles

plot(real(zs2), imag(zs2), 'o', 'MarkerSize', 8, 'MarkerEdgeColor', 'white', 'MarkerFaceColor', 'none', 'LineWidth', 2);

% Plot poles as white crosses

plot(real(ps2), imag(ps2), 'x', 'MarkerSize', 10, 'Color', 'white', 'LineWidth', 3);

grid on;

axis equal;

xlim([-4 2]);

ylim([-4 4]);

xlabel('Real Part');

ylabel('Imaginary Part');

title('H2(s): Poles and Zeros');

legend('Zeros (O)', 'Poles (X)', 'Location', 'best');

hold off;

% Subplot 3

subplot(1,3,3);

hold on;

% Plot zeros as white circles

plot(real(zs3), imag(zs3), 'o', 'MarkerSize', 8, 'MarkerEdgeColor', 'white', 'MarkerFaceColor', 'none', 'LineWidth', 2);

% Plot poles as white crosses

plot(real(ps3), imag(ps3), 'x', 'MarkerSize', 10, 'Color', 'white', 'LineWidth', 3);

grid on;

axis equal;

xlim([-4 2]);

ylim([-4 4]);

xlabel('Real Part');

ylabel('Imaginary Part');

title('H3(s): Poles and Zeros');

legend('Zeros (O)', 'Poles (X)', 'Location', 'best');

hold off;

% Set overall figure title

sgtitle('Pole-Zero Plots for Three Transfer Functions');



Figure 2: Pole-Zero Plots for the Functions

# PART 2: Frequency Response and Bode Plots in MATLAB

## Example: Consider a system function:



1. Define the numerator and denominator polynomial coefficients as vector *b* and *a* respectively.

2. Use the *freqs* function to evaluate the frequency response of a Laplace transform.

H = freqs(b,a,omega);

Where is the frequency vector in rad/s. (Hint: use *linspace* to generate a vector with 200 samples.)

3. Plot the magnitude and phase of the frequency response.

4. Plot the bode plot of the given H(s) by utilizing the results in 2. (Hint: use the definitions of the bode plot)

clear all;

close all;

% 1. Define the numerator and denominator polynomial coefficients

b = [2 2 17]; % Numerator coefficients

a = [1 4 104]; % Denominator coefficients

% Generate frequency vector with 200 samples (rad/s)

omega = linspace(-20, 20, 200); % Frequency range from -20 to 20 rad/s

% 2. Use freqs function to evaluate the frequency response

H = freqs(b, a, omega);

% 3. Plot the magnitude and phase of the frequency response

figure;

% Magnitude plot (linear scale)

subplot(2,2,1);

plot(omega, abs(H), 'b-', 'LineWidth', 2);

grid on;

xlabel('Frequency (rad/s)');

ylabel('|H(jω)|');

title('Magnitude Response (Linear Scale)');

% Phase plot

subplot(2,2,2);

plot(omega, angle(H)\*180/pi, 'r-', 'LineWidth', 2);

grid on;

xlabel('Frequency (rad/s)');

ylabel('Phase (degrees)');

title('Phase Response');

% 4. Plot the Bode plot using results from step 2

% Magnitude in dB vs log frequency

subplot(2,2,3);

semilogx(omega, 20\*log10(abs(H)), 'b-', 'LineWidth', 2);

grid on;

xlabel('Frequency (rad/s)');

ylabel('Magnitude (dB)');

title('Bode Plot - Magnitude');

% Phase in degrees vs log frequency

subplot(2,2,4);

semilogx(omega, angle(H)\*180/pi, 'r-', 'LineWidth', 2);

grid on;

xlabel('Frequency (rad/s)');

ylabel('Phase (degrees)');

title('Bode Plot - Phase');

% Add overall title

sgtitle('Frequency Response Analysis: H(s) = (2s² + 2s + 17)/(s² + 4s + 104)');



Figure 3: Frequency Response Analysis of the Example

## Exercise

### Plot the bode plot of each four system functions given in the part 1.

clear all;

close all;

% Define all transfer functions

b = [1 -1]; % Numerator coefficients - H(s)

a = [1 2 2]; % Denominator coefficients - H(s)

b1 = [1 5]; % Numerator coefficients - H1(s)

a1 = [1 2 3]; % Denominator coefficients - H1(s)

b2 = [2 5 12]; % Numerator coefficients - H2(s)

a2 = [1 2 10]; % Denominator coefficients - H2(s)

b3 = [2 5 12]; % Numerator coefficients - H3(s)

a3 = conv([1 2 10],[1 2]); % Denominator coefficients - H3(s)

% Generate frequency vector with 200 samples (rad/s)

omega = linspace(0.1, 100, 200); % Positive frequencies for Bode plots

% Calculate frequency responses for all transfer functions

H = freqs(b, a, omega);

H1 = freqs(b1, a1, omega);

H2 = freqs(b2, a2, omega);

H3 = freqs(b3, a3, omega);

% Figure 1: H(s) = (s-1)/(s²+2s+2)

figure;

subplot(2,1,1);

semilogx(omega, 20\*log10(abs(H)), 'b-', 'LineWidth', 2);

grid on;

xlabel('Frequency (rad/s)');

ylabel('Magnitude (dB)');

title('H(s) = (s-1)/(s²+2s+2): Bode Magnitude Plot');

subplot(2,1,2);

semilogx(omega, angle(H)\*180/pi, 'b-', 'LineWidth', 2);

grid on;

xlabel('Frequency (rad/s)');

ylabel('Phase (degrees)');

title('H(s) = (s-1)/(s²+2s+2): Bode Phase Plot');

% Figure 2: H1(s) = (s+5)/(s²+2s+3)

figure;

subplot(2,1,1);

semilogx(omega, 20\*log10(abs(H1)), 'r-', 'LineWidth', 2);

grid on;

xlabel('Frequency (rad/s)');

ylabel('Magnitude (dB)');

title('H1(s) = (s+5)/(s²+2s+3): Bode Magnitude Plot');

subplot(2,1,2);

semilogx(omega, angle(H1)\*180/pi, 'r-', 'LineWidth', 2);

grid on;

xlabel('Frequency (rad/s)');

ylabel('Phase (degrees)');

title('H1(s) = (s+5)/(s²+2s+3): Bode Phase Plot');

% Figure 3: H2(s) = (2s²+5s+12)/(s²+2s+10)

figure;

subplot(2,1,1);

semilogx(omega, 20\*log10(abs(H2)), 'g-', 'LineWidth', 2);

grid on;

xlabel('Frequency (rad/s)');

ylabel('Magnitude (dB)');

title('H2(s) = (2s²+5s+12)/(s²+2s+10): Bode Magnitude Plot');

subplot(2,1,2);

semilogx(omega, angle(H2)\*180/pi, 'g-', 'LineWidth', 2);

grid on;

xlabel('Frequency (rad/s)');

ylabel('Phase (degrees)');

title('H2(s) = (2s²+5s+12)/(s²+2s+10): Bode Phase Plot');

% Figure 4: H3(s) = (2s²+5s+12)/((s²+2s+10)(s+2))

figure;

subplot(2,1,1);

semilogx(omega, 20\*log10(abs(H3)), 'm-', 'LineWidth', 2);

grid on;

xlabel('Frequency (rad/s)');

ylabel('Magnitude (dB)');

title('H3(s) = (2s²+5s+12)/((s²+2s+10)(s+2)): Bode Magnitude Plot');

subplot(2,1,2);

semilogx(omega, angle(H3)\*180/pi, 'm-', 'LineWidth', 2);

grid on;

xlabel('Frequency (rad/s)');

ylabel('Phase (degrees)');

title('H3(s) = (2s²+5s+12)/((s²+2s+10)(s+2)): Bode Phase Plot');





Figure 4: Bode Plots for the Four Transfer Functions

### 2. Select three sinusoidal signals with unit magnitude, zero phase and three different frequencies (f1,f2,f3 , here fi = RegistrationNumber \* i). Assume that they are three inputs for abovementioned four systems. Then find the corresponding three outputs for each system.

Registration Number = 420

f1 = 420 Hz

f2 = 840 Hz

f3 = 1260 Hz

clear all;

close all;

% Define symbolic variables

syms s t

% Define all transfer functions symbolically

H0 = (s-1)/(s^2+2\*s+2);

H1 = (s+5)/(s^2+2\*s+3);

H2 = (2\*s^2+5\*s+12)/(s^2+2\*s+10);

H3 = (2\*s^2+5\*s+12)/((s^2+2\*s+10)\*(s+2));

% Base frequency

base\_freq = 420; % kHz

for i = 1:3

f\_kHz = base\_freq \* i; % Frequencies: 420, 840, 1260 kHz

f = f\_kHz \* 1000; % Convert to Hz for calculations

% Laplace transform of input signal sin(2\*pi\*f\*t)

input\_signal\_laplace = laplace(sin(2\*pi\*f\*t), t, s);

% Define time vector for plotting

Fs = 1e8; % Sampling frequency (100 MHz for high resolution)

time = 0:1/Fs:2e-5; % Time vector (from 0 to 20 microseconds)

% Create figure for this frequency

figure;

% Plot the input signal

subplot(5,1,1);

input\_signal = sin(2\*pi\*f\*time); % Input signal in time domain

plot(time\*1e6, input\_signal, 'w-', 'LineWidth', 2);

title(['Input: sin(2π × ', num2str(f\_kHz), ' kHz × t)']);

xlabel('Time (μs)');

ylabel('Amplitude');

xlim([0 20]);

grid on;

% H0(s) = (s-1)/(s²+2s+2)

output\_signal\_laplace = H0 \* input\_signal\_laplace;

output\_signal\_time = ilaplace(output\_signal\_laplace, s, t);

output\_signal\_eval = double(subs(output\_signal\_time, t, time));

subplot(5,1,2);

plot(time\*1e6, output\_signal\_eval, 'b-', 'LineWidth', 2);

title('H0(s) = (s - 1)/(s² + 2s + 2)');

xlabel('Time (μs)');

ylabel('Amplitude');

xlim([0 20]);

grid on;

% H1(s) = (s+5)/(s²+2s+3)

output\_signal\_laplace = H1 \* input\_signal\_laplace;

output\_signal\_time = ilaplace(output\_signal\_laplace, s, t);

output\_signal\_eval = double(subs(output\_signal\_time, t, time));

subplot(5,1,3);

plot(time\*1e6, output\_signal\_eval, 'r-', 'LineWidth', 2);

title('H1(s) = (s + 5)/(s² + 2s + 3)');

xlabel('Time (μs)');

ylabel('Amplitude');

xlim([0 20]);

grid on;

% H2(s) = (2s²+5s+12)/(s²+2s+10)

output\_signal\_laplace = H2 \* input\_signal\_laplace;

output\_signal\_time = ilaplace(output\_signal\_laplace, s, t);

output\_signal\_eval = double(subs(output\_signal\_time, t, time));

subplot(5,1,4);

plot(time\*1e6, output\_signal\_eval, 'g-', 'LineWidth', 2);

title('H2(s) = (2s² + 5s + 12)/(s² + 2s + 10)');

xlabel('Time (μs)');

ylabel('Amplitude');

xlim([0 20]);

grid on;

% H3(s) = (2s²+5s+12)/((s²+2s+10)(s+2))

output\_signal\_laplace = H3 \* input\_signal\_laplace;

output\_signal\_time = ilaplace(output\_signal\_laplace, s, t);

output\_signal\_eval = double(subs(output\_signal\_time, t, time));

subplot(5,1,5);

plot(time\*1e6, output\_signal\_eval, 'm-', 'LineWidth', 2);

title('H3(s) = (2s² + 5s + 12)/((s² + 2s + 10)(s+2))');

xlabel('Time (μs)');

ylabel('Amplitude');

xlim([0 20]);

grid on;

% Add overall title for the figure

sgtitle(['System Responses to sin(2π × ', num2str(f\_kHz), ' kHz × t) Input']);

fprintf('Completed analysis for frequency: %d kHz\n', f\_kHz);

end







Figure 5:Outputs for the 3 Selected Frequencies

# PART 3: Surface Plots of a System Function in MATLAB

## Exercise

### Where are the poles and zeros on the surface plot? What’s the relationship between the surface plot and the plot in 2.(2) ?.

clear all;

close all;

% Define range and number of points for sigma and omega

sigma = linspace(-20, 20, 100);

omega = linspace(-20, 20, 100);

% Create meshgrid for sigma and omega

[sigmagrid, omegagrid] = meshgrid(sigma, omega);

% Combine sigma and omega into a complex plane grid (s-plane)

sgrid = sigmagrid + 1i\*omegagrid;

% Define coefficients for numerator (b) and denominator (a) polynomials

b = [2 2 17];

a = [1 4 104];

% Calculate frequency response H1 using polynomial evaluation

H1 = polyval(b, sgrid)./polyval(a, sgrid);

% Calculate magnitude in dB, handling potential infinities

mag\_dB = 20\*log10(abs(H1));

% Replace any infinite or NaN values for better visualization

mag\_dB(isinf(mag\_dB) | isnan(mag\_dB)) = -100;

% Create a mesh plot

figure;

mesh(sigmagrid, omegagrid, mag\_dB);

xlabel('σ (Real Part)');

ylabel('ω (Imaginary Part)');

zlabel('|H(s)| (dB)');

title('3D Magnitude Response: H(s) = (2s² + 2s + 17)/(s² + 4s + 104)');

colorbar;

grid on;

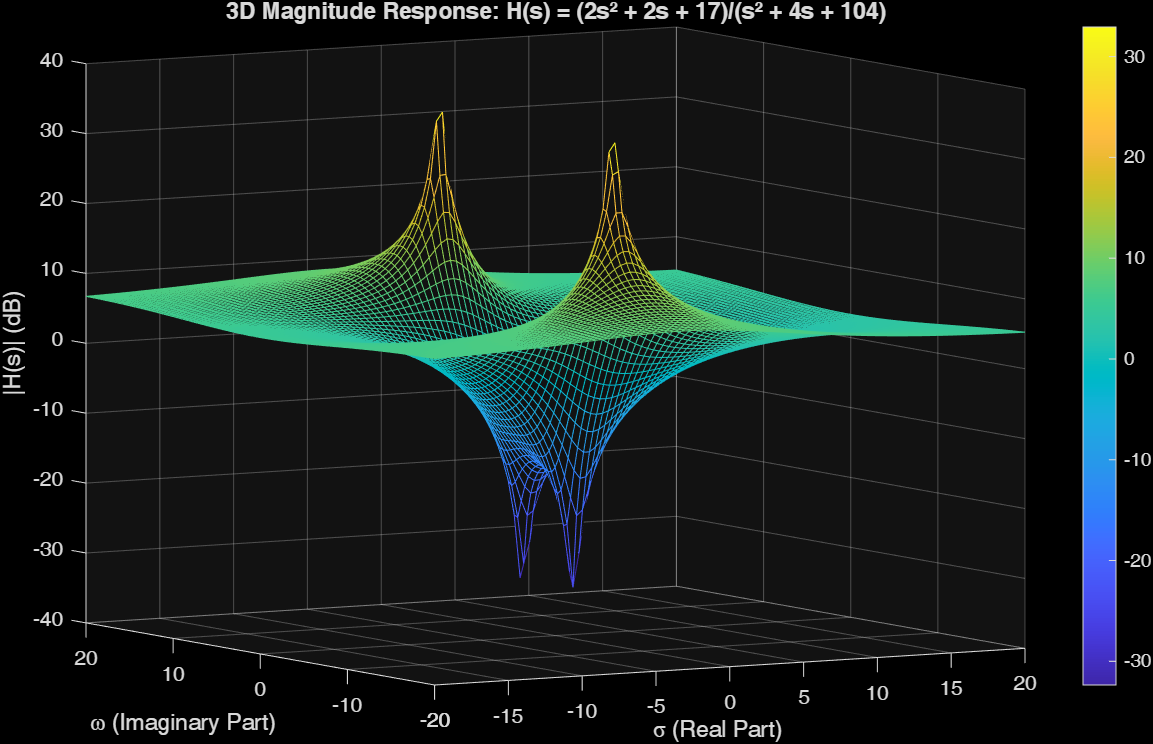


Figure 6:3D Magnitude Plot

**Locations of Zeros and Poles:**

* **Poles:** These are seen as sharp peaks or upward spikes on the surface, where the magnitude approaches infinity.
* **Zeros:** These show up as deep dips or notches, where the surface drops down to zero.

**Connection to the Bode Plot:**

The Bode plot is generated by evaluating the transfer function along the line s = jω, which means setting σ (the real part) to zero. As a result, the Bode magnitude plot represents a cross-sectional view of the 3D surface along the σ = 0 plane, displayed on a logarithmic scale.