EEE342 Spring 2025 Lab 2 Preliminary Work

Emir Arda Bayer

Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey

1. Introduction

The purpose of this preliminary work was to find the Bode plot of the DC motor's transfer function and gain an understanding of this concept of plotting. Later, by the process of applying sinusoidal inputs to the transfer function, a new Bode plot that displays the magnitude and phase difference was created and analyzed.

2. Laboratory Content

2.1 Question 1

Using the code given in the manual on MATLAB, the bode plot of the DC motor's transfer function was found as in Figure 1, without using the 'bode' method of MATLAB.

$$G_p(s) = \frac{20}{0.5s + 1}$$

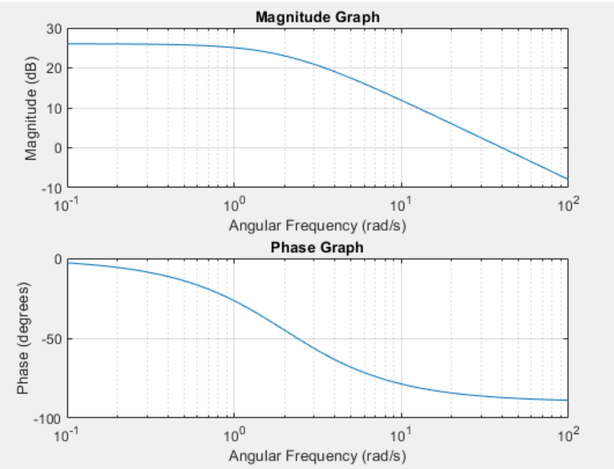

Figure 1: Transfer Function's Bode Plot

2.2 Question 2

Regarding Question 2, the transfer function of the DC motor was inputted a sinusoidal input and a new Bode plot was generated to interpret the sinusoidal output for magnitude and phase differences. This sinusoidal input was inputted with changing 10 different angular frequencies. The following information that is also given in the manual was used in the process.

$$y(t) = A \times |G(jw)| \times \cos(wt + \angle G(jw)) \quad \text{when} \quad x(t) = A \times \cos(wt)$$

The 'fft' method of MATLAB was used while finding the magnitude and phase differences between the input and output signals, and by using 'semilogx' command with 'x' markers a comparison with Question 1 was made possible as these markers were placed on the plot in Q1.

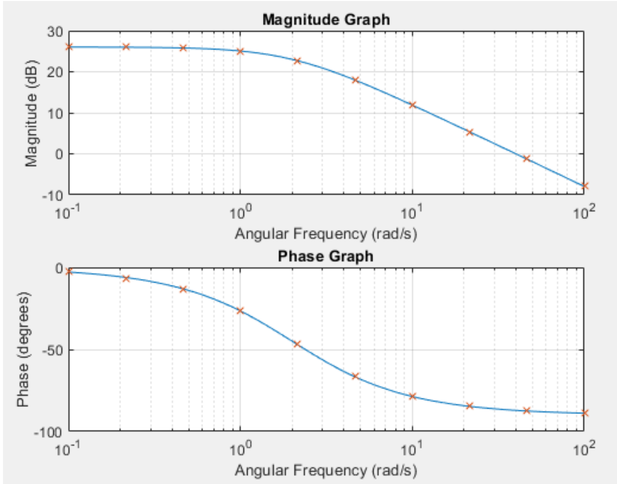

Figure 2: Bode Plot When Applied a Sinusoidal Input

3. Conclusion

For the second preliminary work, the DC motor's transfer function's bode plot was generated manually with no use of MATLAB's 'bode' functionality. Later, a sinusoidal input was applied, the bode plot for the transfer function was again generated and using 'x' markers the magnitude and phase difference were interpreted. In this preliminary work the method to manually generate Bode plots for a given transfer function was observed.

4. MATLAB Code

```
w = logspace(-1,2,100);
for k = 1:100
s = 1i * w(k);
G(k) = 20 / (0.5*s+1);
end
subplot(2,1,1)
semilogx(w,20*log10(abs(G)));
title('Magnitude Graph');
xlabel('Angular Frequency (rad/s)');
ylabel('Magnitude (dB)');
grid on
subplot(2,1,2)
semilogx(w,angle(G)*180/pi)
title('Phase Graph');
xlabel('Angular Frequency (rad/s)');
ylabel('Phase (degrees)');
grid on

% code regarding q2
t = 0:0.01:100;
n = logspace(-1, 2, 10);
for k = 1:10
    s = 1i * n(k);
    G2(k) = 20 / (0.5 * s + 1);
    x = 3 * cos(n(k)*t);
    y = 3 * abs(G2(k)) * cos(n(k)*t + angle(G2(k)));

    X = fft(x);
    [magX,index] = max(abs(X));
    phX = angle(X(index));
    Y = fft(y);
    [magY,index] = max(abs(Y));
    phY = angle(Y(index));

    mag(k) = magY / magX; % final magnitude
    ph(k) = phY - phX; % final phase
end
```

```
subplot(2,1,1)
semilogx(w,20*log10(abs(G)))
hold on
semilogx(n,20*log10(mag), 'x')
title('Magnitude Graph');
xlabel('Angular Frequency (rad/s)');
ylabel('Magnitude (dB)');
grid on

subplot(2,1,2)
semilogx(w,angle(G)*180/pi)
hold on
semilogx(n,ph*180/pi, 'x')
title('Phase Graph');
xlabel('Angular Frequency (rad/s)');
ylabel('Phase (degrees)');
grid on
```