EEE 342 Lab 2 Report – Spring 2025

Emir Arda Bayer

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

1. Introduction

The purpose of this lab is to evaluate a DC motor in the frequency domain in order to perform system identification studies. Inputting sinusoidal signals since the outputs will also be sinusoidal as the system is LTI, with changing magnitudes and phases. There will be numerous signals inputted with varied angular frequencies to conduct system identification.

2. Laboratory Content

Question 1 – Generation of Theoretical Bode Plot

Using the transfer function found in the first lab, DC motor's theoretical bode plot was generated thanks to the code provided in the lab manual.

$$G(s) = \frac{13.75}{0.131s + 1}$$

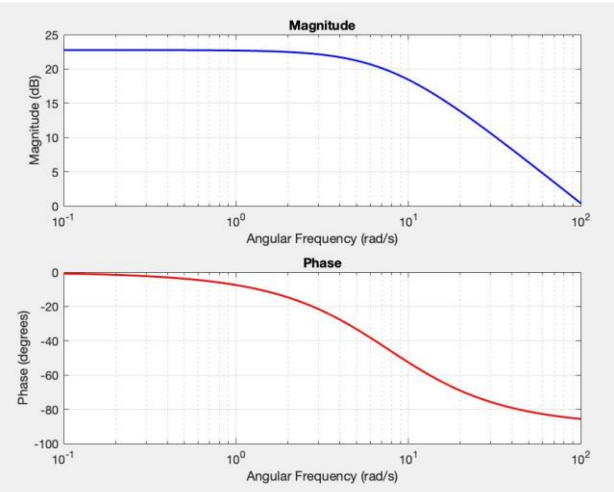

Figure 1: Theoretical Bode Plot

Having obtained this plot, comparisons with changing inputs on hardware will be possible. Frequency domain system identification will help in generating new bode plots.

Question 2 – Generation of Experimental Bode Plots

Regarding the second question, the velocity of the physical DC motor was evaluated by applying varied sinusoidal inputs. By changing angular frequencies and simulation durations as in Table 1, results were obtained.

Angular Frequency (rad/s)	Duration of Simulation (s)
0.1	70
0.3	70
1	25
3	25
10	10
30	10
100	10

Table 1: Varying Inputs to DC Motor

For each combination of angular frequency and simulation duration, the desired values were entered into SIMULINK and output DC motor velocities were saved. As the inputs were generated by the provided code, simulation results were ready to be plotted on top of the input signal. The hardware response for input signal with angular frequency of 0.1 rad/s was as in Figure 2.

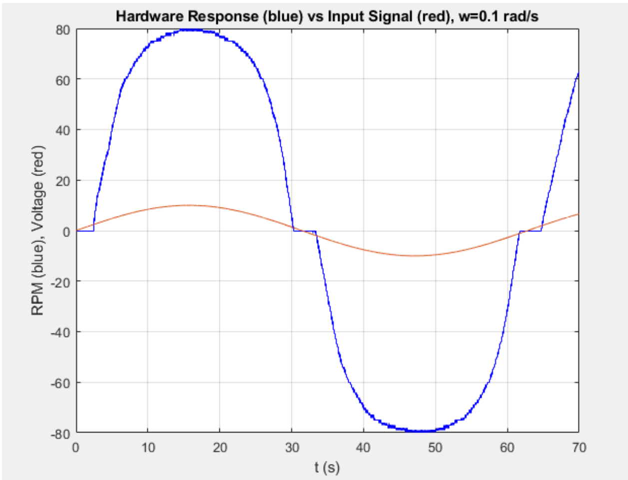

Figure 2: Output Sinusoidal for Input with w = 0.1 rad/s

As the output velocity for the motor was recorded, using FFT (Fast Fourier Transform) new magnitude and phase values were obtained.

$$[maxVel, maxVelIndex] = max(abs(fft(velocity.signals.values)))$$

$$K = \frac{max\,(abs(fft(velocity.signals.values)))}{max\,(abs(fft(input))}$$

The above equations were used to obtain the magnitudes and Matlab's 'abs' and 'fft' commands were used in the process. Similarly, phase of fft signals in input and output were subtracted to find the phase difference as below.

$$\phi_{velocity} = angle(fft(velocity.signals.values))$$

$$\phi_{input} = angle(fft(input))$$

$$\phi = \phi_{velocity}(maxVelIndex) - \phi_{input}(maxVelIndex)$$

As the magnitude and phase for angular frequency w = 0.1 rad/s was recorded, it was added on the initial bode plot as an ‘x’, and new values for varying angular frequencies were also added as new recordings and calculations were made.

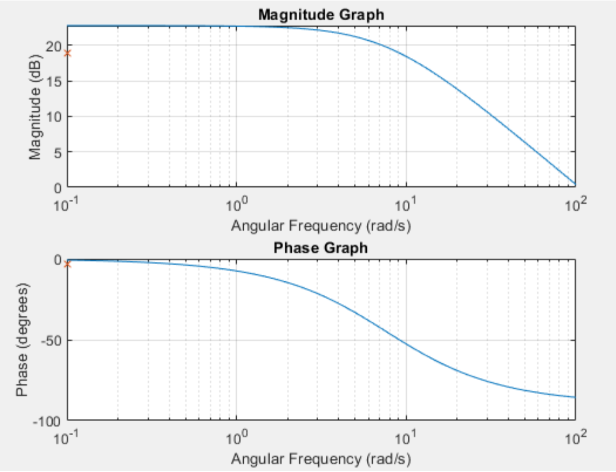

Figure 3: Experimental Values on Initial Bode Plot

The new input output relations of the DC motor for 6 new angular frequencies, and their calculated magnitude and phase values are as in the following figures.

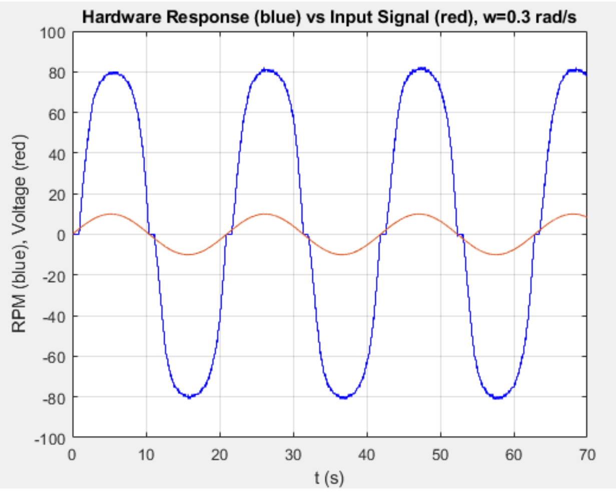

Figure 4: Output Sinusoidal for Input with w = 0.3 rad/s

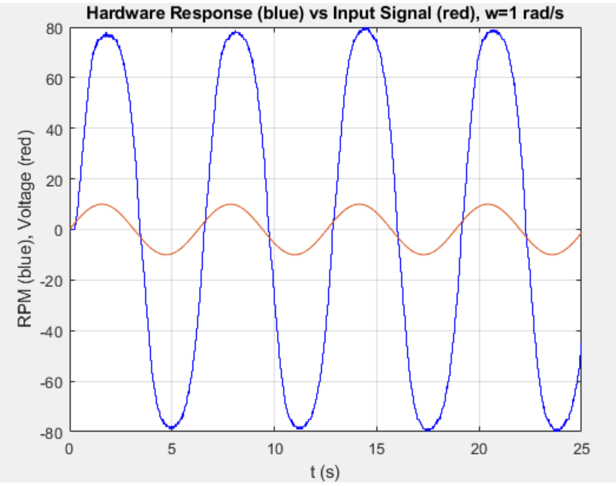

Figure 5: Output Sinusoidal for Input with w = 1 rad/s

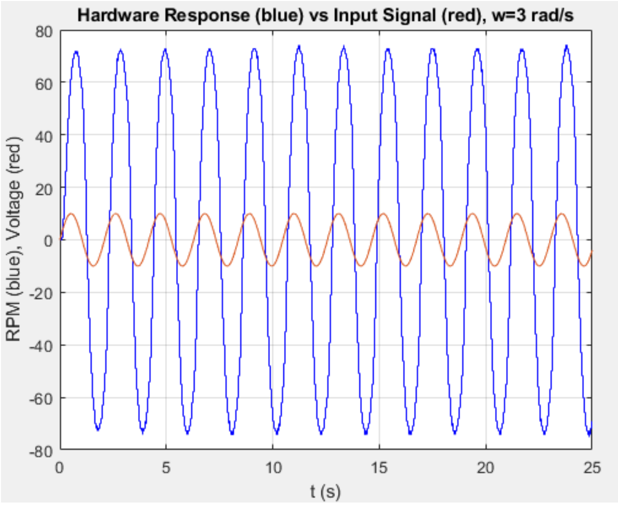

Figure 6: Output Sinusoidal for Input with w = 3 rad/s

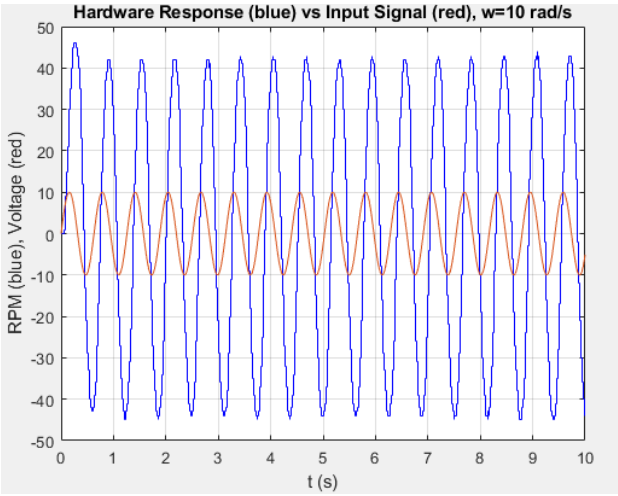

Figure 7: Output Sinusoidal for Input with w = 10 rad/s

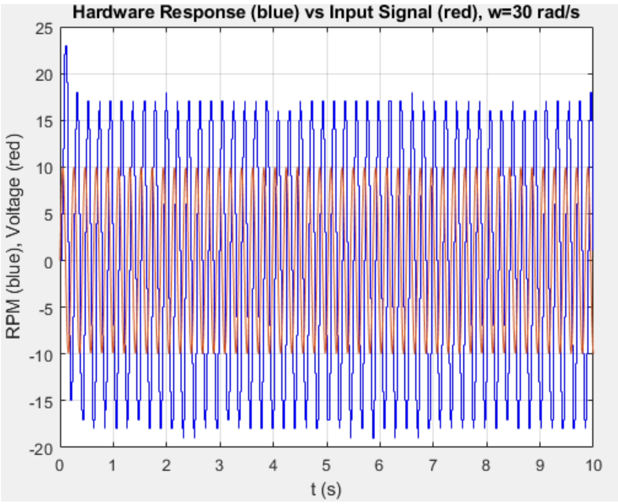

Figure 8: Output Sinusoidal for Input with w = 30 rad/s

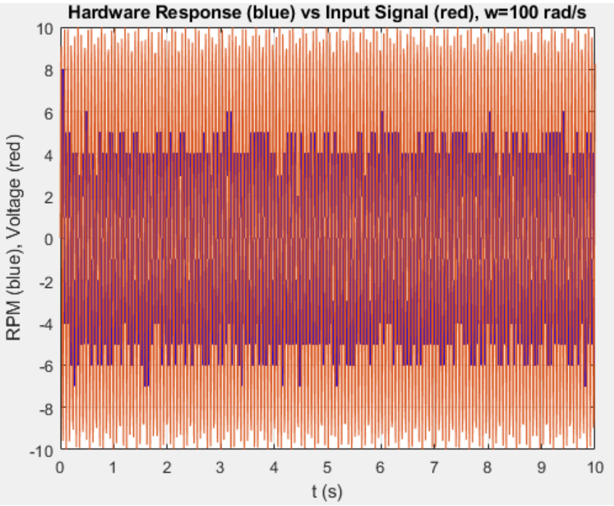

Figure 9: Output Sinusoidal for Input with w = 100 rad/s

As the FFT (Fast Fourier Transform) of input and output signals were calculated by Matlab, each angular frequency's magnitude and phase values were obtained and shown as 'x's on the Figure 10 below.

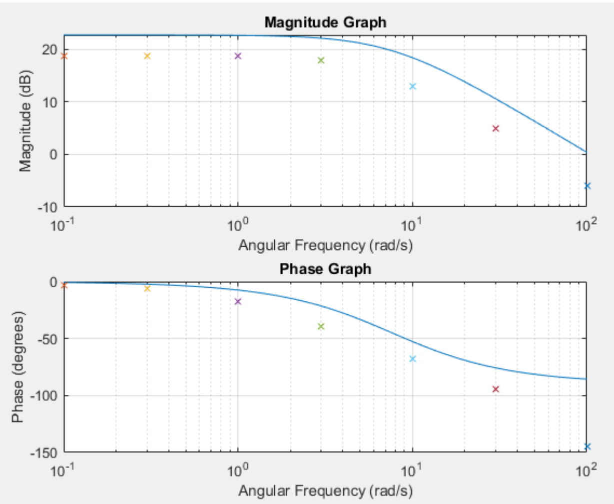

Figure 10: All Experimental Values on Initial Bode Plot

Besides the phase difference for w = 100 rad/s, the final values were similar to the initial bode plot of the DC motor's transfer function.

Question 3 – Time Delay's Effect on Bode Plot

Processing requirements generate a 10 ms delay creating phase differences at high frequencies since this time difference is not taken into account through the estimated transfer function. Pade Approximation was used in order to eliminate this phase difference.

The transfer function was updated to $G_{delayed}$ to consider this 10 ms delay as below.

$$G_{delayed}(s) = G(s)\frac{1 - 0.005s}{1 + 0.005s}$$

Later, an addition of Pade approximation to the plot in Figure 2 was made to evaluate the phase difference error.

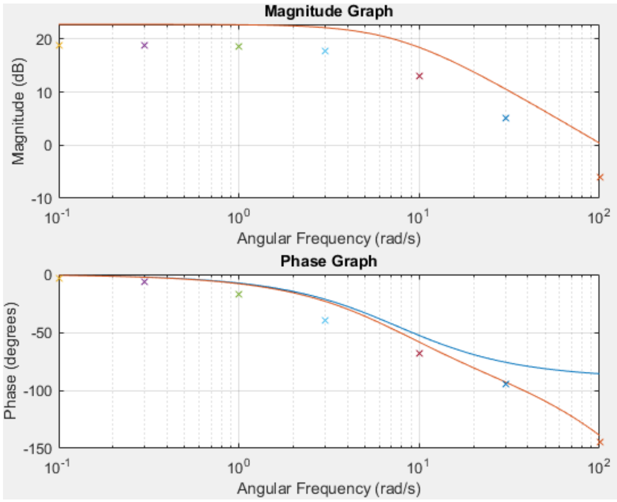

Figure 11: Theoretical, Experimental and Pade Approximation on Bode Plot

The above figure displays how phase difference error at high frequencies caused by processing requirements is eliminated with the use of Pade Approximation and an update of transfer function. This brought theoretical and experimental results closer, validating initial estimates.

3. Conclusion

The purpose of this lab was to conduct system identification studies in the frequency domain on the provided physical DC motor. Varied sinusoidal inputs were applied into motor and output velocities were recorded. Later, applying FFT (Fast Fourier Transform) to inputs and outputs changing magnitude and phase values were obtained to compare and analyze against the initial theoretical transfer function's bode plot. As there was a time delay due to processing requirements, an error in phase differences at high frequencies was evident, and Pade Approximation was used to update the transfer function and eliminate this time related error. This way, the system identification of a DC motor in the frequency domain was achieved.

4. MATLAB Code

```
%% Q1

w = logspace(-1,2,100);
for k = 1:100
s = 1i * w(k);
G(k) = 13.75 / (0.131*s+1);
Gd(k) = G(k)*((1-0.005*s)/(1+0.005*s));
end
subplot(2,1,1)
semilogx(w,20*log10(abs(G)));
grid on
subplot(2,1,2)
semilogx(w,angle(G)*180/pi)
grid on

%% Q2, Q3

%step6
angular_frequency = 100;
duration     = 10;
t = 0:0.01:duration;
input = 10*sin(angular_frequency * t);

vel = out.velocity;
plot(vel, 'b')
hold on
grid on
plot(out.tout, input);
```

```
title('Hardware Response (blue) vs Input Signal (red), w=100 rad/s');
xlabel('t (s)');
ylabel('RPM (blue), Voltage (red)');

%step7
X = fft(input);
[magX,indexX] = max(abs(X));
phX = angle(X(indexX));

Y = fft(vel.Data);
[magY,indexY] = max(abs(Y));
phY = angle(Y(indexY));

k = magY/magX;
ph = phY - phX;
k01=8.7310;
k03=8.7108;
k1=8.5862;
k3=7.7972;
k10=4.4384;
k30=1.7841;
k100=0.5;

ph01=-0.0564;
ph03=-0.1133;
ph1=-0.3018;
ph3=-0.6907;
ph10=-1.1823;
ph30=-1.6453;
ph100=-2.521;

subplot(2,1,1)
semilogx(w,20*log10(abs(G)))
hold on
semilogx(w,20*log10(abs(Gd)))
hold on
semilogx(0.1,20*log10(k01), 'x')
 hold on
semilogx(0.3,20*log10(k03), 'x')
hold on
semilogx(1,20*log10(k1), 'x')
hold on
semilogx(3,20*log10(k3), 'x')
hold on
semilogx(10,20*log10(k10), 'x')
hold on
semilogx(30,20*log10(k30), 'x')
hold on
semilogx(100,20*log10(k100), '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(w,angle(Gd)*180/pi)
hold on
semilogx(0.1, ph01*180/pi, 'x')
hold on
semilogx(0.3, ph03*180/pi, 'x')
hold on
semilogx(1, ph1*180/pi, 'x')
hold on
semilogx(3, ph3*180/pi, 'x')
hold on
semilogx(10, ph10*180/pi, 'x')
hold on
semilogx(30, ph30*180/pi, 'x')
hold on
semilogx(100, ph100*180/pi, 'x')
title('Phase Graph');
xlabel('Angular Frequency (rad/s)');
ylabel('Phase (degrees)');
grid on
```