Lab 1 – System Identification

Ayhan Okuyan

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

1. Introduction

In the lab assignment, we were supposed to integrate on and model the workings of a DC Motor Plant and measure its response to a step input. On the lab, we were provided with a kit that consisted of a DC Motor Plant, an Arduino UNO with a shield to drive it. First, we have configured the feedback system on Arduino, by using MATLAB and Simulink. and then, we have received the data on its motion. We have used the provided lab_1_position_loop_step.slx file to rotate the DC Motor 90 degrees. Also, we have recorded the data that we have gathered from the motion and saved it onto the MATLAB workspace.

Then, we have modelled the same exact model that we have configured by hand, by using the same calculations that we have performed on the Preliminary Lab Work, which will also be discussed in the further sections. Given the LPF (low pass filter) values, we have assumed the second order is dominant and found the ω and ζ values. We have done this based on looking at the data and estimating through maximum overshoot and settling time. Then, we have found the initial DC Motor parameters τ and K and presented them to the model.

Then, we have altered the LPF values according to the ones that we have provided by the lab instructions. Then, we have plotted the Bode plots of the data that we have gathered, whose results will also be discussed throughout the report.

2. Laboratory Content

2.1 Hardware and Configuration

The configuration that we have made is done in order to test the work of the DC Motor and provide connection in between the kit and the MATLAB/Simulink Environment. For this part, we have followed the instructions provided. While doing that, we have investigated the Simulink configuration of the position loop that we have uploaded to the Arduino, and the inner structure of the DC Motor plan block, which are given below.

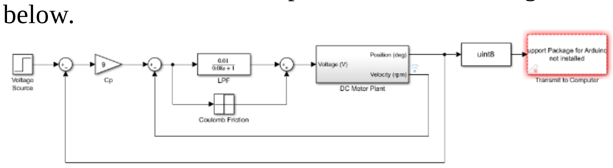


Fig. 1: The Simulink configuration. (Since I have taken the screenshot of the structure on my home PC, Arduino Support Package was not installed at that instant, and I have received an error according to its absence)

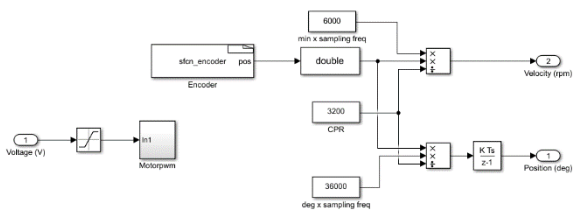


Fig. 2: The Simulink configuration of the DC Motor plant block.

Hence, we have configurated a COMM port for both sending and receiving the data. Then, we have run the code, and observed the behavior by first sight and second, the data that we have taken from the system, which is given below.

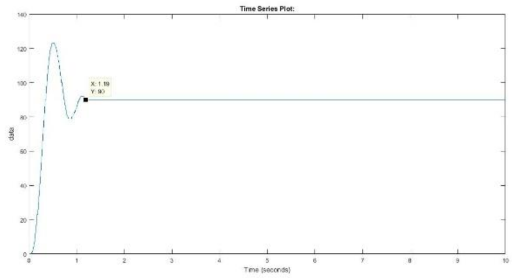
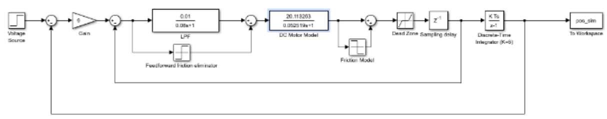


Fig. 3: The data that we have emulated from the system. Then, we moved onto the second part of the lab when we confirmed that the system is functional.

2.2 Time Domain Identification

Then, by using the system configuration that we have considered in Section 2.1, we have modelled the systems given by the instructors. However, in this part, we use a more integrated version of the block diagram which by itself considers the forward and backward frictions. The model is given below.



Although this system looks more complicating, the math that we use is the same as before hence, it would be illuminating to go through the calculations and derivations that we have already passed through in the Preliminary Lab Work 1 and continue with the analysis of the first system we are given.

2.2.1 Mathematics, First System and MATLAB Script

In order to evaluate the ω and ζ, we have used the maximum overshoot and settling time approximations, which are given by

$$\frac{M(0) - y(\infty)}{y(\infty)} = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$$

$$T_{Set} = \frac{4}{\zeta\omega_0}$$

Here, by looking at the data, we observed that the maximum overshoot is equal to 123 which is the $M(0)$ value and the settling time to be 1.19 seconds. Hence, when we plug these values, results become as given below.

$$\frac{123 - 90}{90} = 0.366 = e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}$$

Hence, ζ becomes 0.304223 and then we use the other formula to detect the ω which turns out to be 11.048932 rad/s. The data that we have covered that shows the overshoot and settling time is also given below.

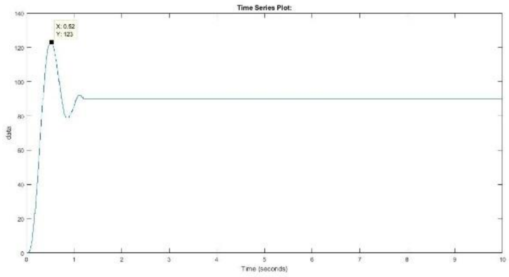


Fig. 4: Data and a Marker for the Max Overshoot

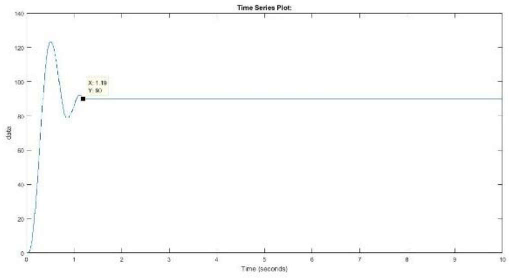


Fig. 5: Data and a Marker for the Settling Time

Furthermore, rather than finding all coefficients by hand, I have reimplemented the MATLAB script that I have written for the Pre-Lab 1. I parametrized all the inputs, even the low pass filter's since they will change throughout the course of this lab assignment. The code first finds the second order characteristic parameters and then the DC Motor coefficients. The code that I have written is given below.

```
syms tau tau_x K zeta w
t_set = 0.69;
m_p = 100;
ss_sol = 90;
k_LPF = 0.1;
tau_LPF = 0.05;
G_c = 2;
G_rpm = 6;

%% Zeta Calc
% Crude Approximation
%eqn_zeta = (m_p-ss_sol)/ss_sol == 1-
(zeta/0.6);
eqn_zeta = (m_p-ss_sol)/ss_sol == exp((-
1*pi*zeta)/sqrt(1-zeta^2));
res_zeta = solve(eqn_zeta, zeta);
fprintf("zeta = %f\n",res_zeta(1));
zeta1 = res_zeta(1);

%% w Calc
eqn_w = t_set == 4/(w*zeta1);
res_w = solve(eqn_w, w);
fprintf("omega = %f\n",res_w(1));
w1 = res_w(1);

%% tau, tau_x, K Calc
eqn1 = (k_LPF*G_c*G_rpm*K)/(1+k_LPF*K) ==
(w1^2)/(2*zeta1*w1 + w1^2*tau_x);
eqn2 = (k_LPF*G_c*G_rpm*K)/(tau_LPF+tau) ==
(w1^2)/(1+2*zeta1*w1*tau_x);
eqn3 = (k_LPF*G_c*G_rpm*K)/(tau_LPF*tau) ==
(w1^2)/(tau_x);
[res_k, res_tau, res_tau_x] =
solve(eqn1,eqn2,eqn3, K, tau, tau_x);
fprintf("K = %f\n",res_k)
fprintf("tau = %f\n",res_tau)
fprintf("tau_x = %f\n",res_tau_x)
```

Then, we use the equations that we have derived from the transfer function characteristics of the parametrized transfer function of the system with the one that we have found. The equations are given below. Remember these calculations are all done via MATLAB.

$$Tr(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}\frac{1}{\tau_x s + 1}$$

$$= \frac{\omega_0^2}{s^3(\tau_x) + s^2(2\zeta\omega_0\tau_x + 1) + s(\tau_x\omega_0^2 + 2\zeta\omega_0) + \omega_0^2}$$

Hence, we find the equations

$$\frac{6K_{LPF}KG_c}{K_{LPF}K} = \frac{\omega_0^2}{\tau_x\omega_0^2 + 2\zeta\omega_0}$$

$$\frac{6K_{LPF}KG_c}{\tau_{LPF} + \tau} = \frac{\omega_0^2}{2\zeta\omega_0\tau_x + 1}$$

$$\frac{K_{LPF}6KG_c}{\tau_{LPF}\tau} = \frac{\omega_0^2}{\tau_x}$$

When we solve these equations for the first system that we are given with LPF coefficients, K_{LPF}= 0.08 and τ_{LPF} = 0.01,and also G_c=9 we get the outputs below

$$\zeta = 0.304223$$
$$\omega_0 = 11.048932$$
$$K = 23.766047$$
$$\tau = 0.054364$$
$$\tau_x = 0.041370$$

We can see the schematics that we obtained by putting in the values below.

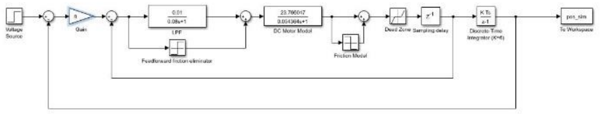


Fig. 6: Closed-Loop System Schematic for the First System

Also the schematics that we upload the data to the Arduino is also given below.

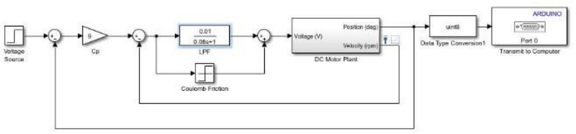


Fig. 7: First System's Load Schematic

Then, we put these values to the Simulink Model that we are given and plot the emulated data and model on top of each other. The graph that we have obtained is given below.

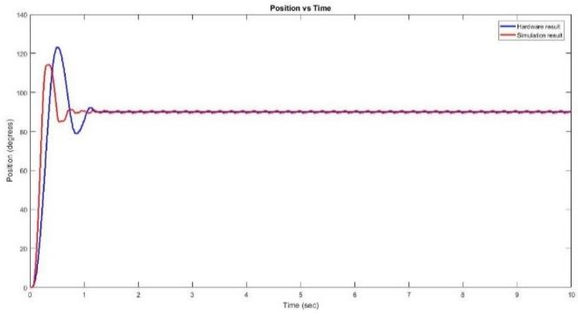


Fig. 8: Results of the Experimentally and Mathematically Obtained Values for the First System

We see that the blue curve represents the data that we have obtained from the model and the red line is the data that we have obtained from the Arduino output. We can see that, with some amounts of delay, the two graphs resemble each other. However, the experimental values overshoot and settling time is relatively small compared to the simulation values. Hence, we conclude the investigation of the first system. The subsequent sections 2.2.2 and 2.2.3 considers the same procedure done on different low pass filter models. However since, the procedure is the same, we only show the findings, results and the further comment on these results.

2.2.2 Second System

For the second system, the values for the controller gain and low pass filter are given as K_{LPF}= 0.1 and τ_{LPF} = 0.6,and also G_c = 2. For these values, we enter them to the system code and get the new data emulated from the motion of the DC Motor. Then by observing the graph we obtained, we find the max overshoot and settling time which are 121rpm and 1.12sec respectively. Furthermore, we run the same calculations that we have gone through in Section 2.2.1 and the results are as given below.

$$\zeta = 0.333051$$
$$\omega_0 = 10.723372$$
$$K = 47.422093$$
$$\tau = 0.031993$$
$$\tau_x = 0.038789$$

The obtained Simulink schematics are given below.

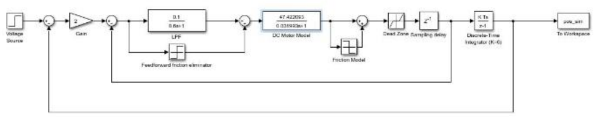


Fig. 9: Closed-Loop System Schematic for the Second System

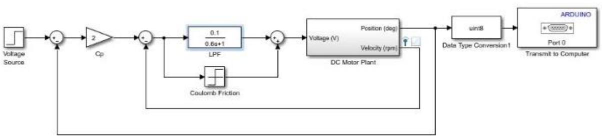


Fig. 10: First System's Load Schematic

Then, we put the coefficients that we have found to our own model and plot both graphs on top of each other.

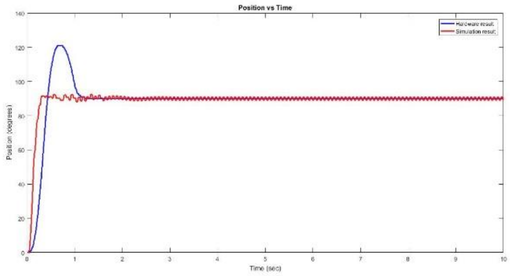


Fig. 11: Results of the Experimentally and Mathematically Obtained Values for the Second System

In this system, we see that the steady state values do match at 90 rpms, which is what was expected. Also, the settling times seems to match with each other. However, the experimental data does not resemble the simulation data in terms of waveform and overshoot. Here, these uncorrelation shows, I believe the effect of linearization. The τ_x and τ values are too close to each other in order to consider one of them insignificant.

2.2.3 Third System

In the third system, the parameters are given as, K_{LPF}= 0.1 and τ_{LPF} = 0.05,and also G_c = 2. Following the same procedure with the Section 2.2.2 with these new values, we first find the overshoot and settling time to be 100rpm and 0.69sec respectively. Then we find the second order and DC Motor coefficients as

$$\zeta = 0.573131$$
$$\omega_0 = 10.114786$$
$$K = 8.998988$$
$$\tau = 0.132173$$
$$\tau_x = 0.062611$$

Again, the obtained schematics are given.

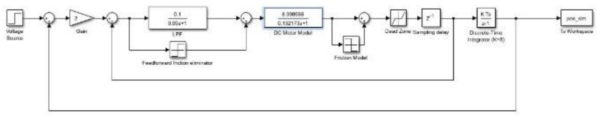


Fig. 12: Closed-Loop System Schematic for the Third System

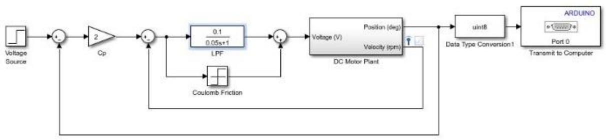


Fig. 13: First System's Load Schematic

Then, we again plot the experimental and simulation data on top of each other. The plot is given below.

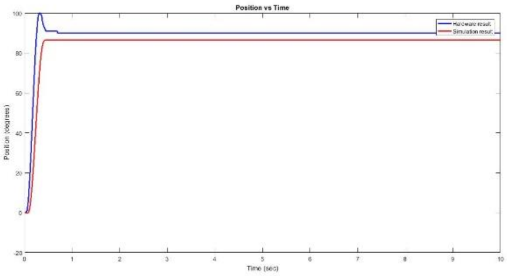


Fig. 14: Results of the Experimentally and Mathematically Obtained Values for the Third System

In this plot, the first thing we notice is that the steady state value between the experimental and simulation data varies a little. As is not significant, the real significance lies more on the waveform. Here we don’t see the overshoot in the simulation, which can represent that we are over-filtering the data. Other than that, the settling time seems to be the indifferent in both cases.

2.3 System Identification, Introduction to Bode Plot

Then, after Section 2 is considered done, we are required to plot the Bode plots of these three different systems on top of each other. The plot is given below.

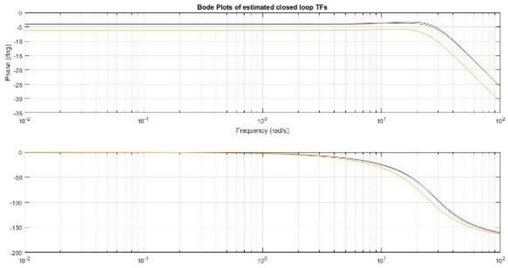


Fig. 15: Bode Plots of the three Systems

We clearly see here that all these systems show low pass filter type effects where the high frequency responses are attenuated.

.

3. Conclusion

In this lab assignment, we have studied three different closed loop feedback systems and their transfer characteristics. We have used a kit that contained a shielded Arduino and a DC Motor by controlling its working principles via MATLAB/Simulink. Considering also the pre-lab, we have first modelled the DC Motor and found the transfer function that had three distinct poles. Then, we assumed a second order system is dominant within the overall system and solved the system according to that. Then, we came to the lab and with given instructions for different parameters, we have run and simulated the workings of the motor and compared the two. Overall, it was a useful assignment to learn and investigate more Simulink and understand further the workings of a closed loop system with a DC Motor in it.

REFERENCES

1. Dorf ,Richard C and Bishop, Robert H. Modern Control Systems. Essex: Pearson. Thirteent edition. (2017)