Feedback Control Systems Lab1 Report

Eray Gündoğdu

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

1. Introduction

The aim of this laboratory work is to investigate input output relations of a DC motor system and making comparison between theoretical and experimental data. In the continuing step a PI controller is designed regarding control of the angular velocity.

2. Laboratory Content

2.1 Hardware Setup

In this part, it is desired to successfully complete the hardware setup by following the steps in the lab manual and plot the received data of the hardware result.

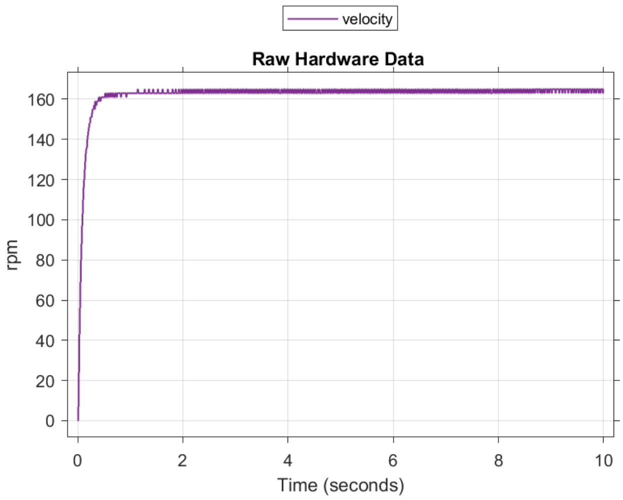

Figure 1 Hardware result of the angular velocity(ω)

The hardware connection is successful and recieved hardware plot of the output is similar to the output of the theoretically approximated transfer function in the preliminary work. Certainly there are differences in steady state gain, settling time due to the different data provided.

2.2 System Identification in Time Domain

The objective is to approximate a transfer function after filtering the data in Part 1 and compare the output of the generated model with the raw data. The LPF which is used in filtering is as following:

$$H(s)_{LPF} = \frac{1}{0.001s + 1}$$

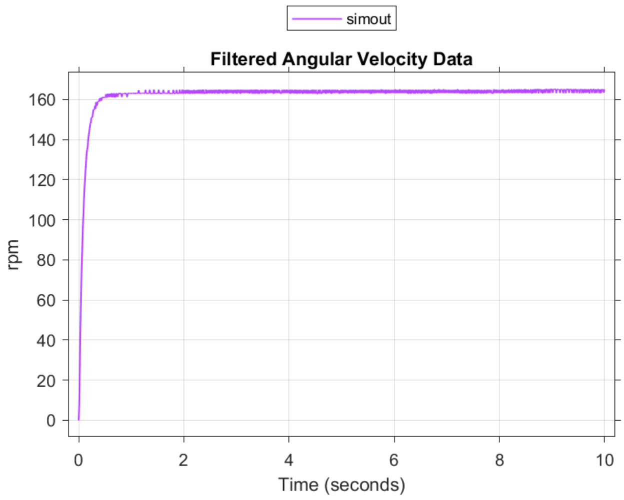

Figure 2 Filtered angular velocity data

It is known that the transfer function is in the form of

$$H(s) = \frac{K_1}{\tau_1 s + 1}$$

In this experiment input is 12V hence the output can be written as

$$Y(s) = H(s)U(s)$$

$$Y(s) = \frac{12}{s} \cdot \frac{K_1}{\tau_1 s + 1}$$

$$y(t) = K_2(1 - e^{\frac{-t}{\tau_1}})$$

By observing the points on Figure 2, K_2 and τ_1 can easily be found.

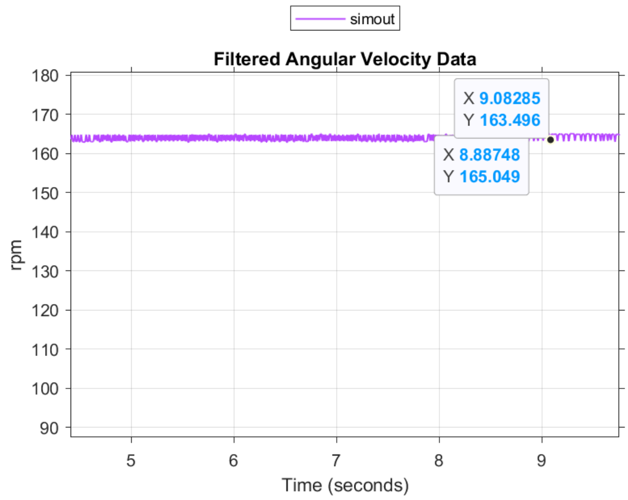

Figure 3 Values for steady state (K_2)

Steady State value is estimated as 164.

$$K_2 \approx 164$$

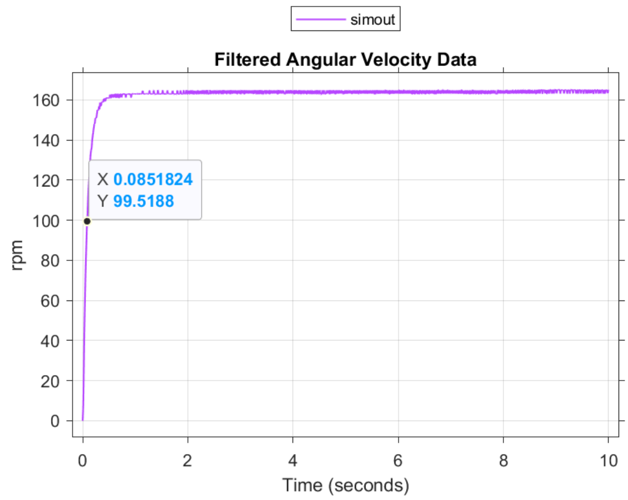

Figure 4 A point for finding time constant

$$99.5188 = K_2(1 - e^{\frac{-0.851824}{\tau_1}})$$

$$\tau_1 \approx 0.091$$

$$K_2 = 12K_1$$

Hence, the approximated transfer function can be expres-sed as:

$$H(s) = \frac{13.6}{0.091s + 1}$$

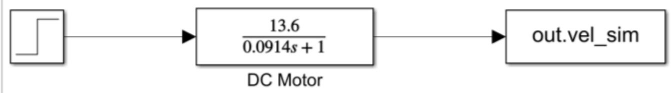

Figure 4 simulation fort he approximated output

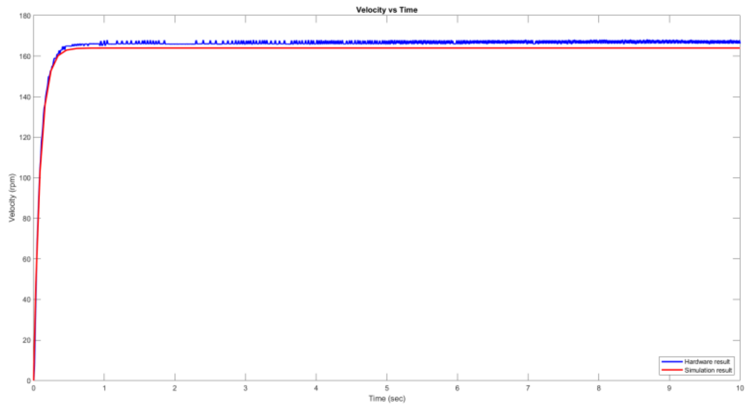

Figure 5 Output of the approximated output vs hardware output

As it can be observed there is slight difference between the plots. In real life, in other words in the hardware result there are oscillations in steady state however in simulation steady state value is written as 164 which is constant. Numerical rounding in determining the time constant may also result in differences. However, as a result two plots are significantly similar to each other.

2.3 PI Controller on Simulation System

A PI controller design is desired which has 0 steady state error, percentage overshoot is less than %8, and settling time is less than 0.8s with %2 error bound. General form of PI controllers in Laplace domain is expressed as[1]

$$G(s) = K_p + \frac{K_i}{s}$$

Initially, K values are assigned to 0.5 in order determine appropriate values for K by trial and error methods.

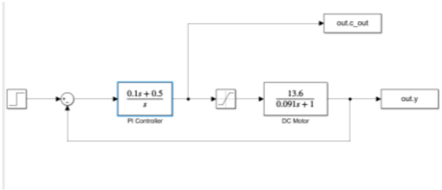

Figure 6 Simulation for the system with PI controller

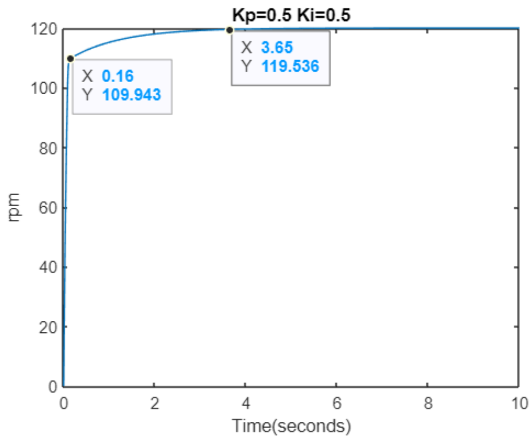

Figure 7 Output when Kp=0.5, Ki=0.5

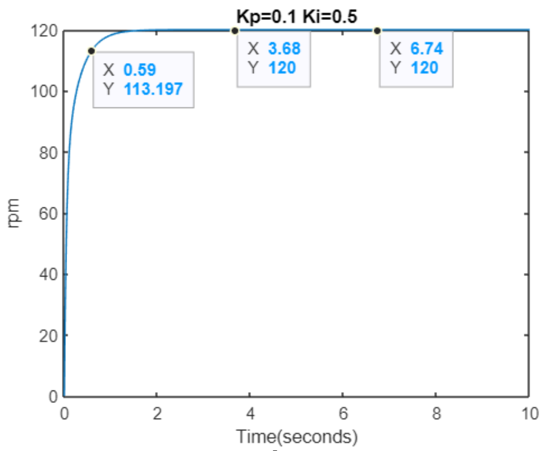

Figure 8 Output when Kp=0.1, Ki=0.5

By observing Figure 8 it can be concluded that the designed PI controller meets the requirements. There is no overshooting and steady state error. In addition, the controller satisfies te settling time which is less than 0.8. Now two different combinations of K pairs which are Kp=10Ki, Ki=20Kp will be applied to the to the PI controller. Further numerical value in terms of the desired specifications is indicated in Table 1.

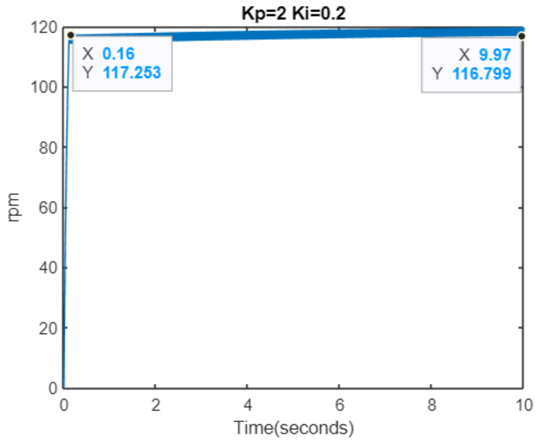

Figure 9 Output when Kp=2, Ki=0.2

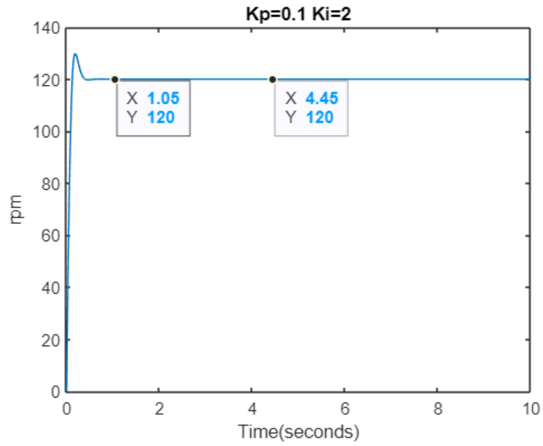

Figure 10 Output when Kp=0.1, Ki=2

	Overshoot(%)	Settling Time(s)	Steady State Error
Kp=0.1 Ki=0.5	0.35	≈ 0.6	0
Kp=2 Ki=0.2	0	>>10	0
Kp=0.1 Ki=2	25	≈ 1.1	0

Table 1.1 Numerical details regarding overshoot percentage, settling time, and steady state error

As it is stated previously the situation where Kp=0.1 and Ki=0.5 suits the requirements hence the specified design is the final model and will be apllied to hardware circuit in the following part.

2.4 PI Controller on Physical System

Now, the designed PI controller is applied to the hardware setup and a comparison between ouput of the designed model and hardware will be done.

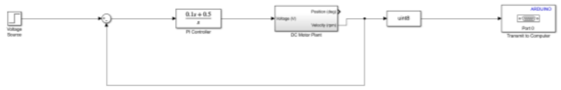

Figure 9 Simulation for hardware connection

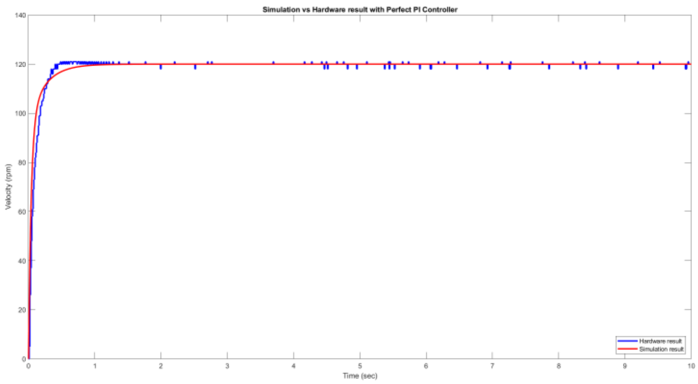

Figure 10 Output of hardware and simulation results with perfect PI controller

Overall the hardware result meets with the requirements in Part 2.3. Additionally, controls the velocity at 120rpm as desired. There are slight differences between simulation and hardware result such as peak time and specified requirements in Part 2.3. This situation is a result of numerical rounding of time constant and the transfer function of DC motor determined with filtered data.

3. Conclusion

The experimental is very useful to familiarize the process of system identification by observing input, output relations and properties such as overshooting, peak time, settling time, steady state error. Moreover, PI controller designing is also prioritized in this lab work. After approximating a transfer function for the DC motor in order to control speed a PI controller is designed which meets the specified requirements in terms of overshooting, steady state error and settling time. The system identification method provided easiness in determining the transfer function parameters o PI controller. After implementing the PI controller to the system by using simulation, hardware result and simulation result of the angular speed is compared. Overall they manage to control the angular speed at 120rpm with similar behavior however there are slight differences between the two plots. Root cause of these slight differences may be a result of, as previously mentioned in Part 2.3, numerical rounding in determining the transfer function parameters and transfer function of the DC motor being determined according to filtered data.

REFERENCES

1. R. C. Dorf and R. H. Bishop, *Modern Control Systems*. Hoboken: Pearson, 2022.

APPENDIX

MATLAB code for PI controllers:

```
plot(out.y)
xlabel('Time(seconds)')
ylabel('rpm')
title('Kp=0.1 Ki=2')
```

MATLAB code for vs plots:

```
figure;
plot(out.velocity,obu,tLineWidth.,2);
hold on; plot(out.y,'r','LineWidth’,2);
xlabel(’Time (sec)’); ylabel(’Velocity
(rpm)’);
title('Simulation vs Hardware result
with Perfect PI Controller');
legend({’Hardware result’,’Simulation
result’},’Location’,’SouthEast’);
```