System Identification

Yusuf Ziya Dilek

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

1. Introduction

This lab consists of 4 parts. First part is preliminary work for setting up the connection between physical system and the software properly. We did not experience any setbacks for this part Matlab-Simulink and physical system both worked as intended. Second part is to find the DC motor's first order approximation just like we did in the preliminary lab report. In the third part, we are tasked with designing PI controller for this DC motor using software tools according to given specifications. At the last part, we tried our designed controller on the physical system itself and plot it together with the simulation results to observe whether simulation and physical system results match with each other.

2. Laboratory Content

Part 1

In this part we followed the instructions closely and set up the software tools properly. It went smoothly without any major trouble.

This Simulink system was given:

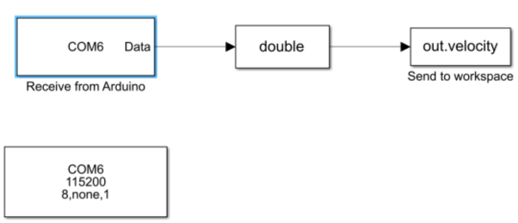

Figure 1. Arduino and DC motor connected to COM6 numbered usb port and received data is written to out timespace

I have saved this reading as vel_1 and moved on to the next stage.

Part 2

In this part we were tasked with identify the given system in time domain after passing it through a low pass filter.

$$H_{lpf} = \frac{1}{0.01s + 1}$$

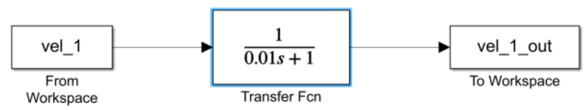

Figure 2.This is the Simulink model for the low pass filtering

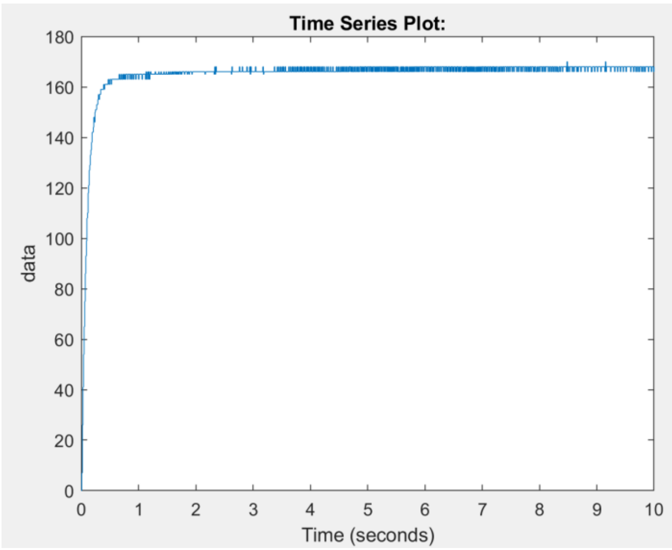

Figure 3. This is before Filtering

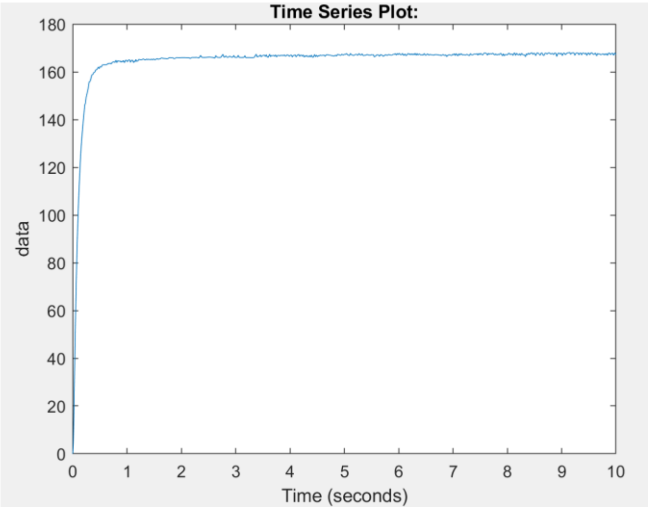

Figure 4. This is after filtering

By using these data, we can estimate the first order estimation of DC motor just like we did in the preliminary work:

$$T(S) = \frac{K}{\tau S + 1} \text{ and } R(s) = \frac{12}{s} \quad then \quad Y(s) = T(s)R(s) = \frac{12K}{s(\tau S + 1)}$$

Inverse Transform:

$$y(t) = 12K(1 - e^{\frac{-t}{\tau}})$$

We know from the plot that:

$$when \quad t \rightarrow \infty \quad y(t) \rightarrow \sim 168 \quad AND \; when \quad t \rightarrow 0.05 \quad y(t) \rightarrow 65$$

By using these two points we can find:

$$\tau = 0.1022 \text{ and } K = 14$$

Then the transfer function becomes:

$$T(S) = \frac{14}{(0.1022)S + 1}$$

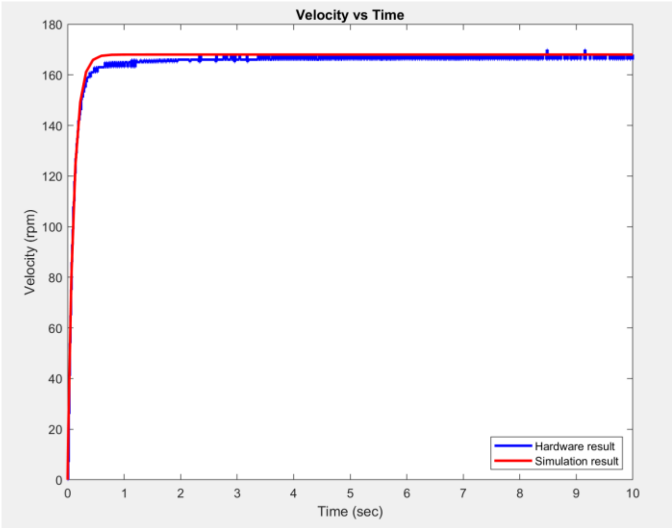

Figure 5. Our estimation vs. the Original

Part 3

Now that we have approximately found a mathematical model of the physical system, we were asked to design PI controllers in computer environment by abiding these parameters:

- Steady state error is 0.
- Maximum percentage overshoot is less than 10%.
- Settling time is less than 0.6 seconds (2% error bound).
- The output of controller does not saturate (i.e. its magnitude does not exceed 12V) after 0.05 seconds.
- The input signal is r(t) = 120u(t) rpm

As stated in the assignment, we started our design with:

CASE 1:

$$K_P = 0.5 \quad AND \quad K_i = 0.5$$

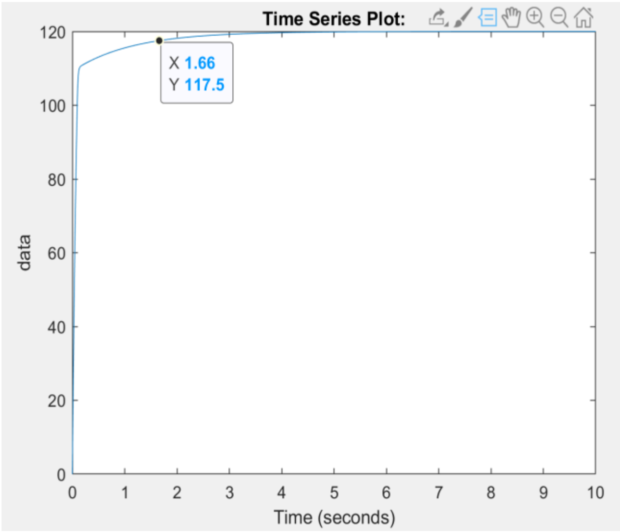

Figure 6. Plot of the case 1

From this we can find that:

$t_s = 1.66 \quad and \quad e_{ss} \approx 0 \quad and \quad PO = 0\%$
Settling time is too long.

CASE 2:

$K_P = 5 \quad AND \quad K_i = 0.5$

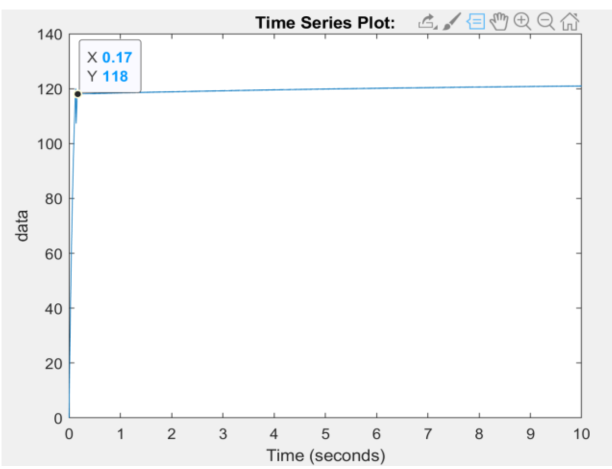

Figure 7. Plot of the case 2

From this we can find that:

$t_s = 0.17 \quad and \quad e_{ss} \approx 0 \quad and \quad PO = 0\%$ Settling time is okay here but output of the controller saturates.

CASE 3:

$K_P = 0.5 \quad AND \quad K_i = 25$

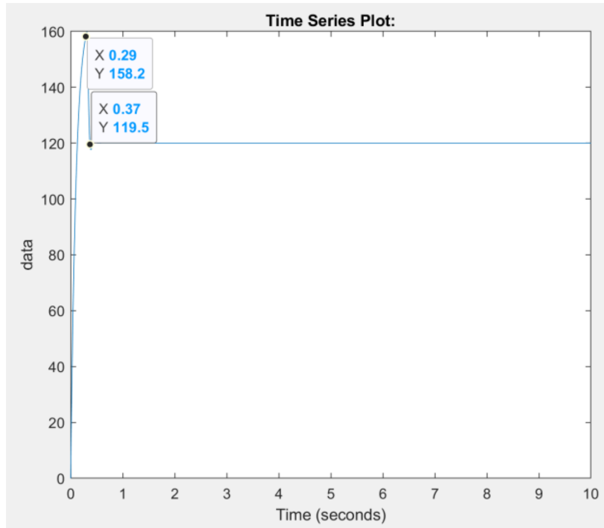

Figure 8. Plot of the third case(very huge overshoot)

From this we can find that:

$t_s = 0.37 \quad and \quad e_{ss} \approx 0 \quad and \quad PO \approx 33\%$
Overshoot is too much.

After trial and error procedure as instructed in the lab assignment we found this case to be reasonable all around:

CASE 4

$K_P = 0.12 \quad AND \quad K_i = 1.5$

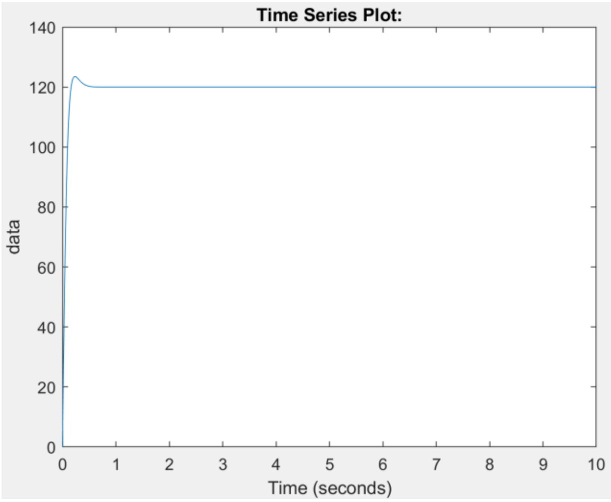

Figure 9. Plot of the case 4

From this we can find that:

$t_s = 0.14 \quad and \quad e_{ss} \approx 0 \quad and \quad PO \approx 2.5\%$ All of this in the expected range.

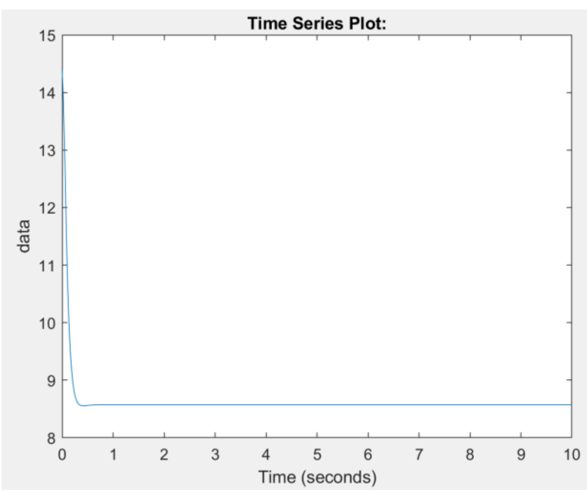

Figure 10. Plot of the output of the controller

As you can see from the figure 10, the output of the controller does not saturate. It only takes around 0.05 seconds to drop to below 12V values.

<u>Part 4</u>

Our simulation results vs. the hardware results in one plot can be seen here:

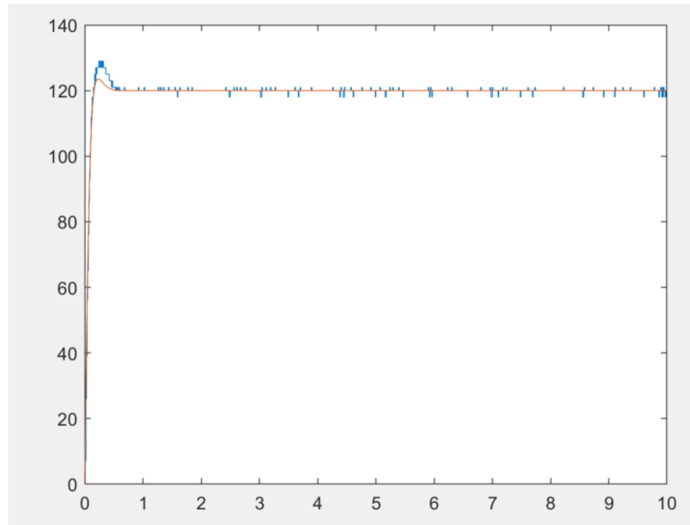

Figure 11. Red is the simulation result and blue is the hardware result

3. Conclusion

In the part 1, software (Matlab-Simulink) and hardware (Arduino-DC motor) worked as expected. We did not face with any major problem aside from the DC motor’s connections was mixed up when we first got our hardware kit but our TAs found 2 mixed cables then we changed connection of those cables with each other and it worked perfectly.

In the part 2, the estimation of first order transfer function was very close to the real world data. The error rate was minimal as expected.

In the part 3, in here we played with coefficients and found a good set of Kp and Ki values that satisfies the given parameters. Finding pairs with good settling time and PO was easy but we needed to play little bit more to bring the saturation time down to 0.05 seconds. Finally in the last part we plotted our simulation results against the hardware results and observed them to be very close.