EEE342 LAB 1:
Approximation and Design of a PI Controller for an Electric Motor

Efe Tarhan

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

1. Introduction

Electrical or mechanical systems are usually does not work as expected or the calculations done for theoretical models does not match with the real-life situations because of the effects of disturbance and uncertainty of the systems. Disturbance is the additional affects from the environment to the system that are not desired or intentionally placed as an input whereas uncertainty is the margin of error when using certain parameters for a system since it is impossible to know all parameters for a real-life system. Feedback systems can be used to overcome the unexpected results or errors coming out of a system and it is the main idea behind this lab work.

Electrical motor being used in this lab converts electrical energy to mechanical energy and therefore gets affected from many unknown effects. Also, it is not an easy task to control this device with an open-loop system where the voltage going through the motor is changed manually by observing the output rpm (revolutions per minute). So, main goal of this lab is to make an approximation for the transfer function of an electrical motor and design a PI controller that controls the motor in the desired speed and under some desired specifications.

Since the motor is type of a system that has many sources of uncertainty and disturbances the output of the hardware system will first inserted to a LPF filter with a pole at s = -1000 then a 1st order approximation will be done to it to be able to design a PI controller with the desired specifications. The model of the PI control is the following:

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

After designing the controller, the next step will be to find the total transfer function of the closed loop system as:

$$H(s) = \frac{G_p(s) \cdot G_c(s)}{1 + G_p(s) \cdot G_c(s)}$$

Where $G_p(s)$ is the approximated function of the electric motor. By using this H(s), the specifications of the system like the maximum overshoot and the settling time will be checked if they are satisfied. All these calculations and measurements will be done on the Chapter 2 of the report. The final remarks and conclusions about the lab will be given in Chapter 3.

2. Laboratory Content

Part 1)

At the first part of the laboratory the raw hardware data has been taken from the motor by using input as

$$r(t) = 12\,u(t)$$

for 10 seconds. The recorded data can be observed in Fig.1.

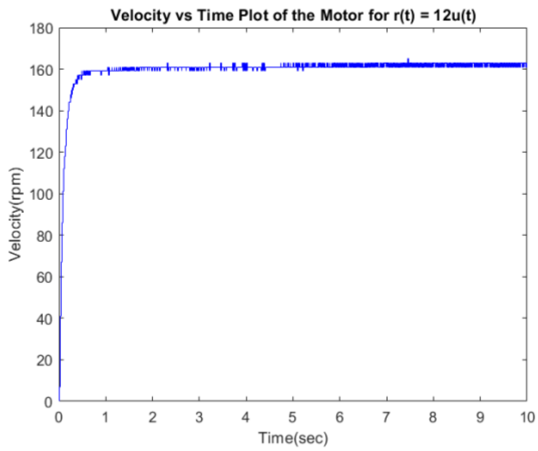

Fig. 1: Velocity-Time Plot of the Motor for r(t) = 12u(t)

If we call the raw data $y_{raw}(t)$ than it can be stated that

$$\lim_{t\to\infty} y_{raw}(t) \approx 160$$

This means that 12V input voltage can create nearly 160 rpm steady state output for this motor. Also, the settling time of the output is less than 0.5 seconds. Important thing to notice here is the presence of the oscillations that are not damped or ended after a while therefore the system is not actually well-controlled by just giving it a voltage level but there is also a need for a feedback system. Reason of these oscillations or noise can be the uncertainty which can be imagined as the unknown physical parameters of the motor and the disturbance effects which are the noise sources in electronical circuits, connections or the inner structure of the motor. In Part 2 the raw output data will be further investigated.

Part 2)

The raw data output of the motor for the input r(t) = 12u(t) has been obtained in Part 1. In this part the output will firstly be filtered by using a LPF with the following form:

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

The filtering operation has been done in order to obtain a good approximation of the transfer function of the motor as a 1st order expression. The obtained filtered result can be seen in Fig.2 with a selected point that the approximation will be done.

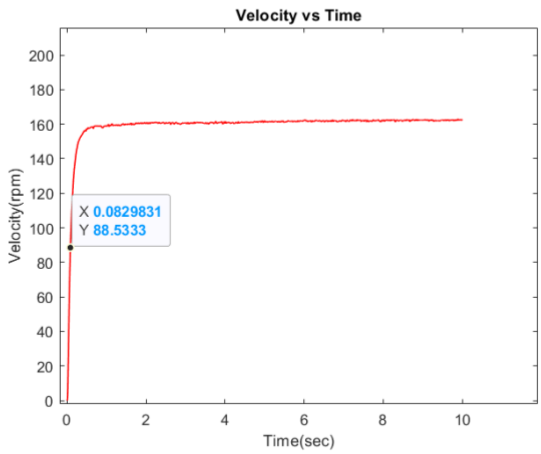

Fig. 2: Filtered Velocity-Time Plot of the Motor for r(t) = 12u(t)

For the first order approximation part the output can be written in the form of

$$y_{appx}(t) = K \cdot (1 - e^{-at})$$

To find the values of this first order approximation the first step is finding the value of K which represents the value of the steady state output at infinity. For this purpose, an output segment between $t \in [2,10]$ has been taken and the average is found which is equal to 162. So, the new approximation becomes the following.

$$y_{appx}(t) = 162 \cdot (1 - e^{-at})$$

Then what can be done is taking an arbitrary point from the graph where the effect of the exponential is visible and then finding a corresponding value of a. For this lab the value can be taken for $t \in [0,0.1]$ which represents a good segment for observing the effects of the exponential component.

The arbitrary selected value is $t = 0.083$ and the corresponding output is equal to the following:

$$y_{raw}(0.083) = 88.533$$

Then the value of a can be found by solving the following equation:

$$162 \cdot (1 - e^{-0.083a}) = 88.533$$

$$e^{-0.083a} = 1 - \frac{88.533}{162}$$

$$a = \frac{\ln\left(1 - \frac{88.533}{162}\right)}{-0.083} \cong 9.5$$

Therefore, the output can be modelled and approximated with the following relation:

$$y_{appx}(t) = 162 \cdot (\, 1 - e^{-9.5t})$$

Since this is the time domain output the transfer function for the motor $G_p(s)$ can be found with the following calculations:

$$Y_{appx}(s) = 162\left(\frac{1}{s} - \frac{1}{s + 9.5}\right) = \frac{162 \cdot 9.5}{s(s + 9.5)}$$

$$R(s) = \frac{12}{s}$$

$$G_p(s) = \frac{162 \cdot 9.5}{s(s + 9.5)} \cdot \frac{s}{12} = \frac{162 \cdot 9.5}{12} \cdot \frac{1}{s + 9.5} = \frac{128.25}{s + 9.5}$$

Therefore, the first order approximation of the system is the found $G_p(s)$.

The transfer function $G_p(s)$ is fed to the Simulink simulation design. The resultant simulation output and the hardware output are recorded on top of each other, and it can be observed in Fig.3.

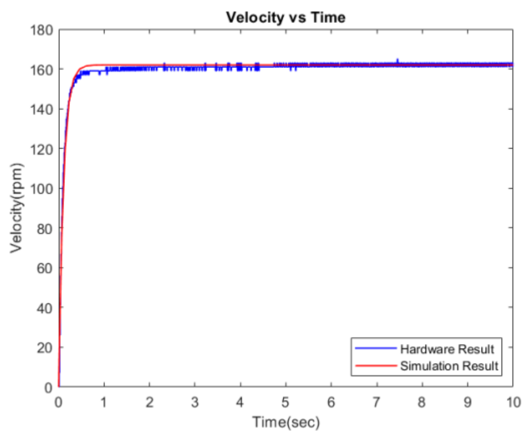

Fig. 3: Velocity-Time Plots of the Motor for the Hardware and the Simulation Outputs

The oscillations are removed as it can be seen from the graph if the hardware and simulation results are compared since the approximation is 1st order. Also, the maximum overshoot value is increased compared to the original hardware data. These are the general results from the approximation of $G_p(s)$. In the Part 3 of the report a PI controller will be designed to control the motor by following the specifications.

Part 3)

In this part three different PI controllers will be proposed to control the motor. From this part let,

$$G_p(s) = \frac{a}{s+b}$$

And

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

Where a = 128.25 and b = 9.5

$$T(s) = \frac{G_c \cdot G_p}{1 + G_c \cdot G_p} = \frac{a(sK_p + K_i)}{s^2 + s(b + aK_p) + K_i a}$$

A) <u>1st PI w/($e_{ss} = 0, T_s < 0.8\ s\ , PO_M < \%8$)</u>

1- $\lim_{t\to\infty} e_{ss}(t) \ = 0$:

By FVT,

$$\lim_{s\to 0} E_{ss}(s)s = sR(s)(1 - T(s))$$

$$\lim_{s\to 0} s \cdot \frac{c}{s} \cdot \frac{s^2 + bs}{s^2 + s(b + aK_p) + K_i a} = 0$$

Therefore, it is already satisfied by choosing.

$$K_i \neq 0$$

2- $PO_M < \%8$:

$$PO_M = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} < 0.08$$

$$\frac{\zeta}{\sqrt{1-\zeta^2}} > \frac{\ln(12.5)}{\pi}$$

By using a calculator:

$$\zeta > 0.6266$$

3- $T_s < 0.8$:

$$\frac{4}{\omega_n \cdot \zeta} < 0.8$$

By using the previous inequality of ζ, the range of ω_n can be found as the following:

$$\omega_n > 7.9796$$

Then,

$$\omega_n^2 = K_i a$$

$$K_i > 0.496$$

Let choose $K_i = 1$ and as a result $\omega_n = 11.32$. If the value of ζ is chosen as 1 which satisfies the inequalities, K_p becomes the following:

$$K_p = \frac{2\zeta\omega_n - b}{a}$$

Then,

$$K_p = 0.1$$

The pair $(K_i, K_p) = (1{,}0.1)$ for the first PI controller design. Result of the simulation of the approximated transfer function and the controller is the can be seen in Fig.4.

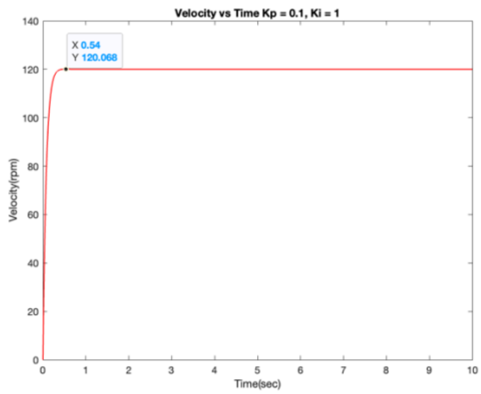

Fig. 4: Velocity-Time Plot of the Motor Simulation for $K_p = 0.1$ and $K_i = 1$.

As it can be seen from the Fig.4 the settling time is less than 0.5 seconds, steady state error is 0 and the percent overshoot can be found to be the following:

$$PO_M = \frac{0.068}{120} = 0.05\%$$

So, all the specifications are set.

B) <u>2nd PI w/($K_p = 10K_i$)</u>

The values of the K_p and K_i are chosen arbitrarily as 1 and 0.1 respectively. The result of the simulation with these controller parameters can be seen in Fig.5

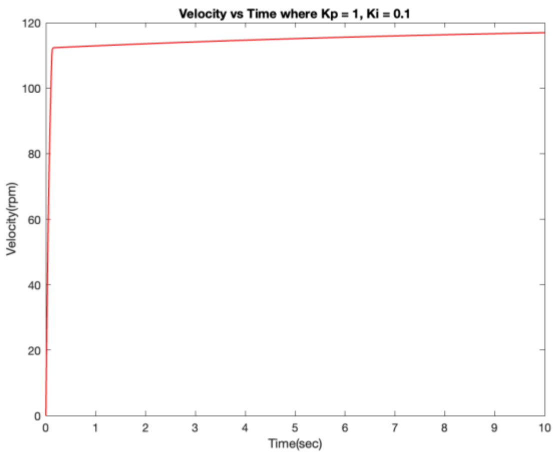

Fig. 5: Velocity-Time Plot of the Motor Simulation for $K_p = 1$ and $K_i = 0.1$.

The designs will be compared at the end by inspecting all three of them.

C) 3rd PI w/ ($10K_p = K_i$)

The values of the K_p and K_i are chosen arbitrarily as 0.1 and 1 respectively. The result of the simulation with these controller parameters can be seen in Fig.6

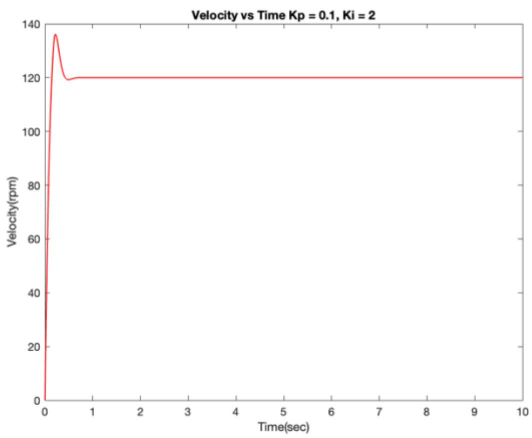

Fig. 6: Velocity-Time Plot of the Motor Simulation for $K_p = 1$ and $K_i = 0.1$.

By comparing all three designs it can be argued that the first solution meets the criteria better than the second and the third since the second solution is undershooted and it couldn't reach 120 rpm even until t = 10 seconds. The third solution has a quite maximum overshoot value and therefore the best solution is the 1st design. The specifications can again be checked from Fig. 7.

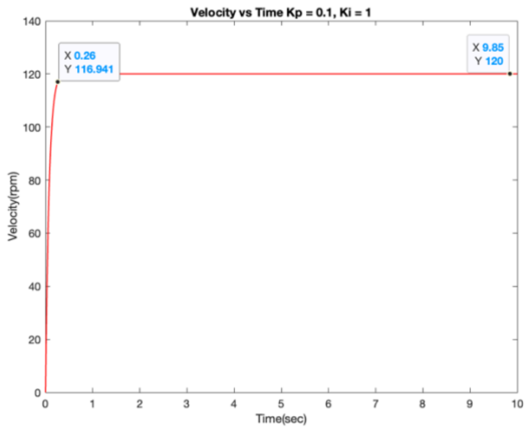

Fig. 7: Velocity-Time Plot of the Motor Simulation for $K_p = 0.1$ and $K_i = 1$.

The steady state error is zero and the settling time is less than 0.2 seconds. Also, the maximum percent overshoot is small enough that it is negligible which has been found in the previous page. In Part 4. This controller design will be used for hardware implementation and the results will be observed.

Part 4)

The controller with the given parameters $K_p = 0.1$ and $K_i = 1$ is used for hardware implementation and the results can be seen from the Fig.8.

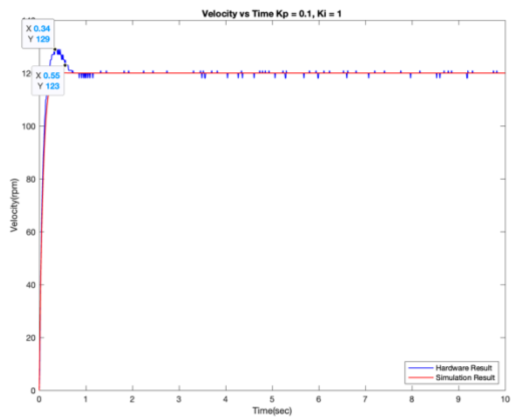

Fig. 8: Velocity-Time Plot of the Motor Hardware Implementation for $K_p = 0.1$ and $K_i = 1$.

Although the expected result was less overshooted, the hardware has a comparably higher overshoot value because of the uncertainty of the system. Also, the same problem creates several oscillations that are not damped through infinity. This problem could have been solved by designing a better approximation, a 2nd or a 3rd of the system and solve accordingly. But still the settling time of the system is 0.56 seconds which satisfies the criteria and maximum percent overshoot is the following:

$$PO_M = \frac{9}{120} = 7.5\%$$

This is also an acceptable result. Identification of the system and usage of the simulation tools of MATLAB allowed me to choose the best results before implementing it on the hardware set up and therefore proved to be useful since not all designs can be tested on a hardware set up in real life problems. Identification of the system by proposing a 1st order approximation made the calculations simpler while the simulations are used for comparing and determining expected results for each design.

3. Conclusion

Aim of this lab is understanding and modelling the mathematical nature of a hardware system like a motor by doing some approximations and designing a controller by using this approximation that allows the output to meet some given specifications.

The sources of error in this lab are the numerical approximations done while using the calculator, uncertainty of the electrical motor component and additional noises or disturbance to the system from outside. But despite all error sources the final design proved to be satisfying all conditions.

This lab work was useful in a way that it solidified the conceptual or theoretical concepts of feedback control systems and showed how to design one using a real life set up.

4. MATLAB Code

```
figure(1)
plot(vel_1,'b','LineWidth',1);
hold on
%%
plot(fitvel_1,"r",'LineWidth',1);
xlabel("Time(sec)");
ylabel("Velocity(rpm)");
title("Velocity vs Time");
legend({"Hardware Result","Filtered
Result"},"Location","Southeast");
%%
infout = round(mean(fitvel_1.Data(10
9:536)));
% y(t) = K(1-e^(-at))
% K = 162
% a = 9.5
%%
figure(2)
plot(vel_1,'b','LineWidth',1);
hold on
plot(vel_sim,"r",'LineWidth',1);
xlabel("Time(sec)");
ylabel("Velocity(rpm)");
title("Velocity vs Time");
legend({"Hardware Result","Simulatio
n Result"},"Location","Southeast");

%%

figure(3)
plot(out1,'r','LineWidth',1);
xlabel("Time(sec)");
ylabel("Velocity(rpm)");
title("Velocity vs Time Kp = 0.1, Ki
 = 1");

figure(4)
plot(out2,'r','LineWidth',1);
xlabel("Time(sec)");
ylabel("Velocity(rpm)");
title("Velocity vs Time where Kp = 1
, Ki = 0.1");

figure(5)
plot(out3,'r','LineWidth',1);
xlabel("Time(sec)");
ylabel("Velocity(rpm)");
title("Velocity vs Time Kp = 0.1, Ki
 = 2");

%%
figure(6)
plot(tout,hardout,'b','LineWidth',1)
;

hold on

plot(out1,'r','LineWidth',1);
xlabel("Time(sec)");
ylabel("Velocity(rpm)");
title("Velocity vs Time Kp = 0.1, Ki
 = 1");
legend({"Hardware Result","Simulatio
n Result"},"Location","Southeast");
```