To: Professor Qi Lu

From: Christopher Jin, Frederic Liu, Gordon Lou, Owen Chen, Yanjun He

Subject: ME1042 Lab 04 Fundamentals of Feedback Control

Date: June 21, 2023

On 14^{th} October, Gordon Lou, Owen Chen, Frederic Liu, Yanjun He, and I conducted our fourth experiment in the course Mechanical Measurements 2, from which we have studied the speed control of a motor with different types of controls and investigated the effect of proportional controls and PI controls on the steady state error.

Proportional Integral controller is a type of controller formed by combining proportional and integral control action (Zhou, 2018). The combination of two different controllers–k_p and k_i produces a more efficient controller which eliminates the disadvantages associated with each one of them. In this case, the control signal shows proportionality with both the error signal as well as with integral of the error signal (Jia et al., 2020).

A control system usually includes the plant, a mathematical model of the system and the controller. Its objective is to make the output of the controlled object reach the desired response. One control method is to use feedback control theory, which produces an error signal by comparing the difference between the output and the controlled signal. This error signal is then sent to the controller to manipulate the input of the controlled object to achieve the goals. There are several benefits of feedback control system like making the controlled object more stable, regulating the output to follow the desired reference, better transient performance, and better disturbance rejection.

A graphical representation of a feedback system (closed-loop system) is called a block diagram as shown in Figure 5, where $R(s)$ is the desired signal, $E(s)$ is the error signal, $U(s)$ is the control signal, $G_c(s)$ is the controller, and $G(s)$ is the plant. In the block diagram, the error signal is $E(s) = R(s) - Y(s)$, the control signal is $U(s) = G_c(s)E(s)$, and the plant's output is $Y(s) = G(s)U(s)$. And all the signals are in Laplace domain. Also, the relationships between these signals are shown in Equation 1 and 2. The final value theorem (FVT) as shown in Equation 3 is applied to check whether the controller design achieves the objective.

In the first part of experiment, we will find the time constant of the feedback system. In the section of System Response, time constant τ of the servo-motor assembly is measured using the time constant tool after running circuit 2.2 file. The recorded graph is shown in Figure 1. In this figure, we obtained that $y_{max} = 6.163$ where $t_1 = 16.365$ s and $y_{min} = 4.045$ where $t_0 = 15.806$ s. We take the time corresponding to the 0.632 segment of the two height differences and finally conclude that our time constant is 0.559 s.

Next, we tune PI control parameters for speed control of a DC motor. Because of the existence of inertia, the speed of DC motor is not always what we desire. There is some delay if we want to change the speed of DC motor. As we increase k_p, the **peak time decreases**, i.e. faster response, **steady state error decreases**. and **overshoot increases** as shown in Figure 2, 3, and 4. From Equation 4, 5, 6, and 7, we can also know that when k_p increases, e_{ss} decreases, t_p decreases, and PO increases.Therefore, we can know that the proportional term of the PI controller adds or subtracts from desired value based on the size of controller error at each time. As error grows or shrinks, the amount added to desired value grows or shrinks immediately and proportionately. The past history and current trajectory of the controller error have no influence on the proportional term computation (Gu et al., 2020). In addition, as we increase k_i, **steady-state error** improves, i.e. **decreases**, response becomes faster, i.e. **peak time decreases**, and **overshoot increases** as shown in Figure 2, 3, and 4, from which we can know that while the proportional term considers the current size of error only at the time of the controller calculation, the integral term considers the history of the error, or how long and how far the measured process variable has been from the set point over time. Integration is a continual summing. Integration of error over time means that we sum up the complete controller error history up to the present time, starting from when the controller was first switched to automatic (Schaum et al., 2020).

The parameter setting of PI controller is the core content of control system design. It determines the proportional coefficient and integral time of the PI controller according to the characteristics of the controlled process. There are many ways to set PI controller parameters, which can be summarized into two categories. The first category is the theoretical calculation setting method. It is mainly based on the mathematical model of the system to determine the controller parameters through theoretical calculations. The calculated data obtained by this method may not be directly usable, and must be adjusted and modified through actual engineering. The second category is the engineering setting method, which mainly relies on engineering experience, is directly carried out in the test of the control system, and the method is simple and easy to master, and is widely used in engineering practice. The engineering setting methods of PI controller parameters mainly include critical proportion method, response curve method and attenuation method. The three methods have their own characteristics. The common point is that they pass the test, and then set the controller parameters according to the engineering experience formula. However, no matter which method is used, the controller parameters obtained need to be adjusted and perfected in actual operation. The general principles of PI debugging are: when the output is not oscillating, increase the proportional gain k_p; when the output is not oscillating, increase the integral gain k_i. The general steps of PI debugging are as follows: The first step is to determine the proportional gain k_p. When determining the proportional gain P, first remove the integral term of PI to make PI a purely proportional adjustment. The input is set to 60% 70% of the maximum value allowed by the system. Gradually increase the proportional gain k_p from 0 until the system oscillates; then conversely, the proportional gain k_p at this time gradually decreases until the system oscillation disappears, record For the proportional gain P at this time, set the proportional gain k_p of PI to 60% 70% of the current value. The adjustment of the proportional gain P is completed. The second step is to determine the integral gain k_i. After the proportional gain P is determined, set a larger initial value of the integral gain k_i, and then gradually decrease k_i until the system oscillates, and then in turn, gradually increase k_i until the system oscillation disappears. Record the k_i at this time and set the PI integral time constant k_i to be 150% 180% of the current value. The integration time constant k_i debugging is completed.

After this experiment, we have learned a lot about the PI controller of DC motor, which promotes the theories we have learned in class. In the future, if we have the opportunity to research from automation in manufacturing, to automotive systems, autonomous systems and even aerospace (Adib Murad et al., 2020), today's experiment will provide us with tremendous help regarding to this usefulness and significance.

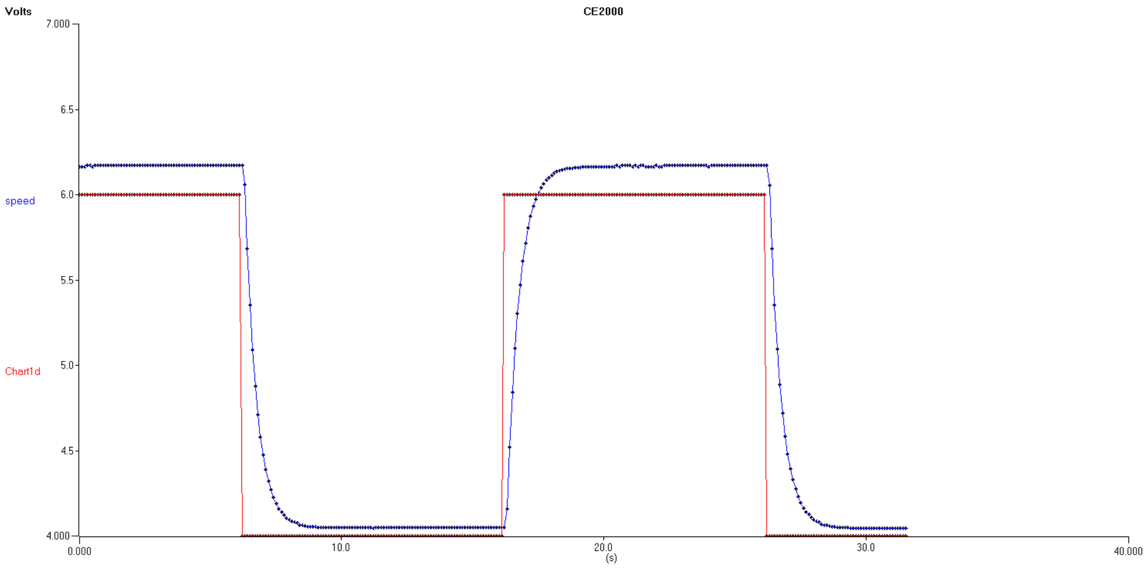

Figure 1: Response with $k_p = 1$.

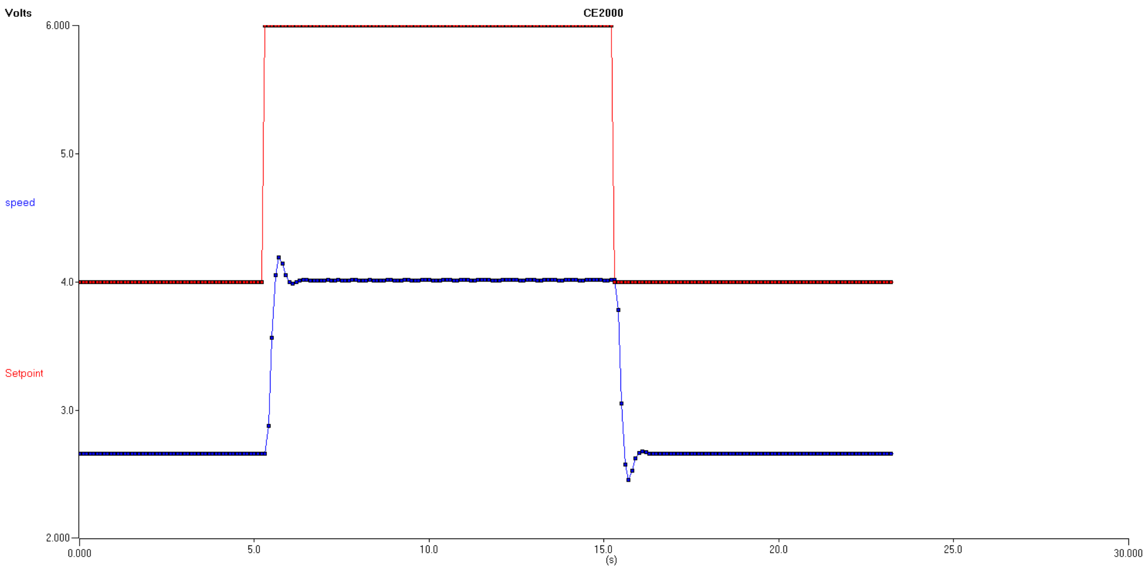

Figure 2: Response with $k_p = 2$.

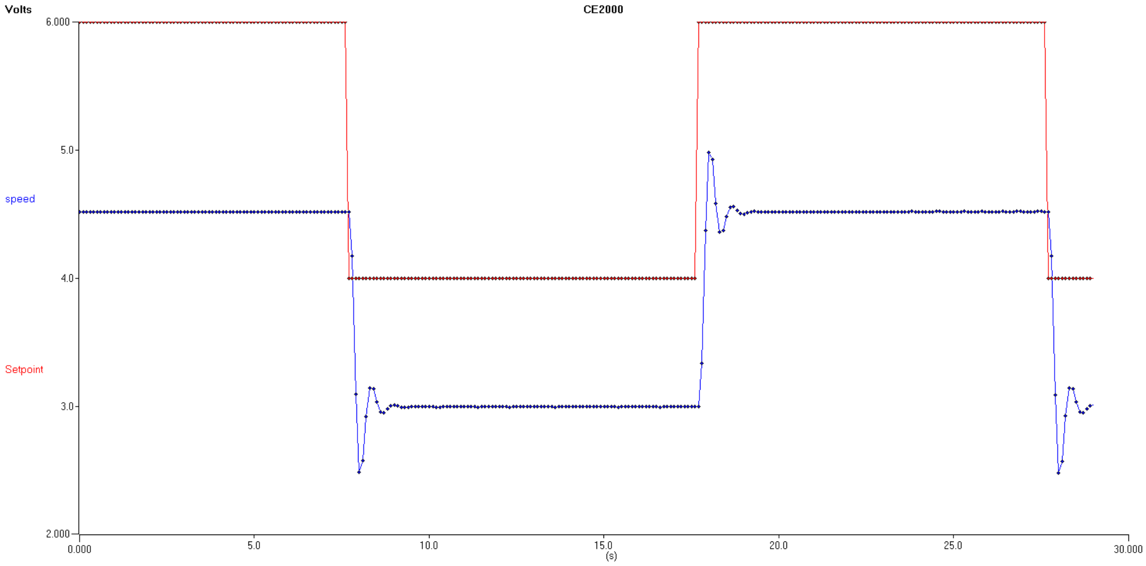

Figure 3: Response with $k_p = 3$.

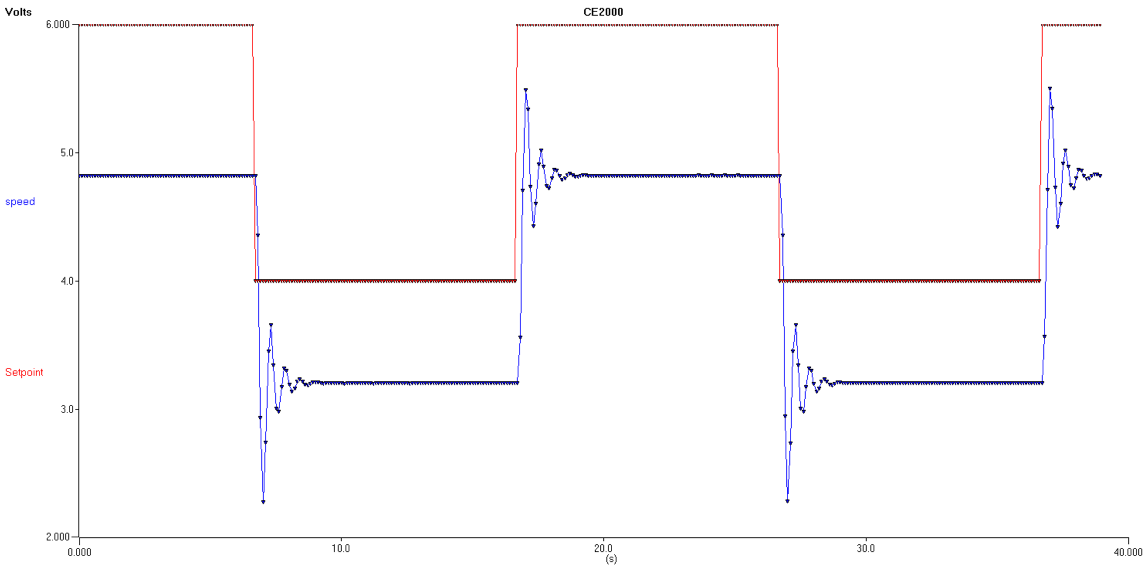

Figure 4: Response with $k_p = 4$.

Table 1: Motor Drive Calibration.

Motor Drive Voltage (V) (Positive)	Motor Speed (rpm)	Motor Drive Voltage (V) (Negative)	Motor Speed (rpm)
0	0	0	0
Dead-Zone Size= 0.2	0	Dead-Zone Size= -0.2	0
1	175	1	-188
2	385	2	-400
3	594	3	-609
4	808	4	-821
5	1022	5	-1034
6	1235	6	-1249
7	1448	7	-1461
8	1661	8	-1675
9	1873	9	-1887
10	$-$	10	$-$

Table 2: Speed Sensor Calibration.

Motor Drive Voltage (V) (Positive)	Speed Sensor Output (V)	Motor Drive Voltage (V) (Negative)	Speed Sensor Output (V)
1	0.88	1	-0.90
2	1.93	2	-1.96
3	2.97	3	-3.00
4	4.03	4	-4.05
5	5.09	5	-5.11
6	6.16	6	-6.18
7	7.22	7	-7.24
8	8.27	8	-8.29
9	9.32	9	-9.35

Table 3: Steady State Error for Various Reference Speeds.

Potentiometer Setting (Reference speed Y_r) (V)	Measured Steady State Error Signal (V)	Theoretical Steady State Error Signal (V)	Speed (V)
2	0.52	0.4893	1.48
3	0.77	0.7340	2.23
4	1.01	0.9787	2.99
5	1.24	1.2233	3.76
6	1.49	1.4680	4.51
7	1.72	1.7127	5.28
8	1.97	1.9574	6.03
9	2.20	2.2020	6.80
10	2.45	2.4467	7.55

Table 4: Steady State Error for Various Controller Gains.

Controller Gain	Measured Steady State Error Signal (V)	Theoretical Steady State Error Signal (V)
2	2.000	1.6350
3	1.484	1.2233
4	1.176	0.9773

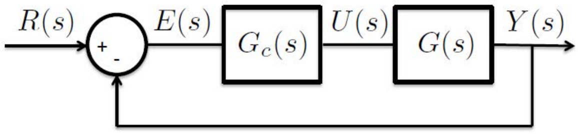

Figure 5: A block diagram of a feedback control system.

Equation for relationship in block diagram:

$$Y\left(s\right)=\frac{G\left(s\right)G_c\left(s\right)}{1+G\left(s\right)G_c\left(s\right)}R\left(s\right) \tag{1}$$

$$E\left(s\right)=\frac{1}{1+G\left(s\right)G_c\left(s\right)}R\left(s\right) \tag{2}$$

Equation for final value theorem:

$$y\left(\infty\right)=\lim_{t\to\infty}y\left(t\right)=\lim_{s\to 0}sY\left(s\right) \tag{3}$$

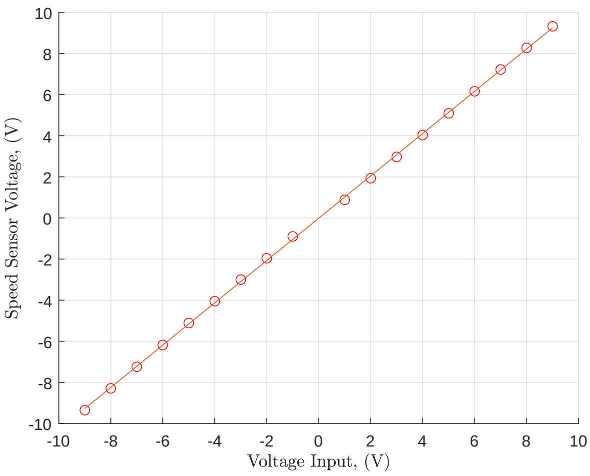

Figure 6: Plot of Voltage Input vs. Speed Sensor Voltage.

Equation for theoretical steady state error:

$$e_{ss}=\frac{Y_r}{1+k_pG} \tag{4}$$

where Y_r is reference speed, k_p is the control gain and G is the calibration gain. **Equation for Expected Peak Time:**

$$t_p=\frac{\pi}{\omega_n\sqrt{1-\xi^2}} \tag{5}$$

Equation for Expected Percent Overshoot:

$$PO=100e^{-\frac{\pi\xi}{\sqrt{1-\xi^2}}} \tag{6}$$

Equation for Proportional Control Gains:

$$k_p=\frac{-1+2\xi\omega_0\tau}{K} \tag{7}$$

Equation for Integral Control Gains:

$$k_i=\frac{\omega^2\tau}{K} \tag{8}$$

Transfer Function Representing the DC Motor Speed-Voltage Relation with Steady-State Gain K and Time Constant τ:

$$\frac{Y(s)}{U(s)}=\frac{K}{\tau s+1} \tag{9}$$

Equation for Input-Output Relation in the Time-Domain for a PI Controller with Set-Point Weighting:

$$u=k_p(b_{sp}\omega_d-\omega_m)+\frac{k_i(\omega_d-\omega_m)}{s} \tag{10}$$

Closed Loop Transfer Function from the Speed Reference ω_d to the Angular Motor Speed Output ω_m:

$$G_{\omega_m,\omega_d}(s)=\frac{K(k_pb_{sp}s+k_i)}{\tau s^2+(Kk_p+1)s+Kk_i} \tag{11}$$

References

Adib Murad, M. A., Tzounas, G., & Milano, F. (2020). Modeling and simulation of fractional order pi control limiters for power systems**this work is supported by the science foundation ireland, by funding mohammed ahsan adib murad, georgios tzounas, and federico milano, under investigator programme grant no. sfi/15/ia/3074. *IFAC-PapersOnLine*, 53(2), 13107–13112. 21th IFAC World Congress.

Gu, D., Yao, Y., Zhang, D. M., Cui, Y. B., & Zeng, F. Q. (2020). Matlab/simulink based modeling and simulation of fuzzy pi control for pmsm. *Procedia Computer Science*, 166, 195–199. Proceedings of the 3rd International Conference on Mechatronics and Intelligent Robotics (ICMIR-2019).

Jia, Y., Chai, T., Wang, H., & Su, C.-Y. (2020). A signal compensation based cascaded pi control for an industrial heat exchange system. *Control Engineering Practice*, 98, 104372.

Schaum, A., Feketa, P., & Meurer, T. (2020). Dissipative pi control for a class of semilinear heat equations with actuator disturbance. *IFAC-PapersOnLine*, 53(2), 7503–7508. 21th IFAC World Congress.

Zhou, J. (2018). Adaptive pi control of bottom hole pressure during oil well drilling. *IFAC-PapersOnLine*, 51(4), 166–171. 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018.