EE302
FEEDBACK SYSTEMS
BONUS PROJECT
FINAL REPORT

Group: Three Musketeers

Huzeyfe Hintoglu – 2093920

Fatih Çalış – 2094845

Recep Günay – 2093888

Introduction

In this project, we are going to design a robot arm controlled by negative feedback using PID control technique. We used Arduino Uno as the controller and L298N module as the motor driver. For the negative feedback, we used a low friction POT which is coupled with the shaft of the DC motor.

We are going to determine certain closed loop system parameters which are necessary for the modeling of the system. For this purpose, we are going to vary K and desired angle values and observe the response of the system via the graphical user interface provided to us. In this report, we are supposed to conduct the first two steps of the project.

Step One

In this step, we have made the necessary connections between Arduino, motor driver and the POT. We observed movement of the arm to different angle values on the GUI. We observed the underdamped response of the system for different parameters. When the Ki and the Kd are zero and Kp value is increased, we observed a faster response with higher overshoot and vice versa.

Step Two

In this step, we inspected the underdamped response of the system to determine the parameters β and ɣ of the system whose block diagram is shown in Figure1.

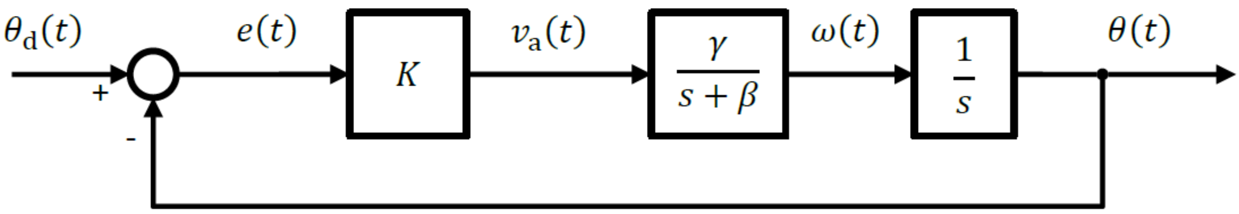

Figure 1: The block diagram of the overall system.

The output Θ(t) appears to be:

$$\theta(t) = \theta_0 + A\left(1 - \frac{\omega_n}{\omega_d} e^{-\sigma t}\cos(\omega_d t - \phi)\right), \qquad t \geq 0\,.$$

The proof of this derivation is as follows:

$$TF(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} = \frac{\omega_n^2}{s^2 + 2\sigma s + \omega_n^2}$$

$$\theta_d(t) = \theta_0 + Au(t) \Rightarrow \theta_d(s) = \frac{\theta_0}{s} + \frac{A}{s} = \frac{\theta_0 + A}{s}$$

$$\theta(s) = TF(s)\,\theta_d(s) = \frac{\omega_n^2(\theta_0 + A)}{s(s^2 + 2\sigma s + \omega_n^2)} = \frac{\omega_n^2(\theta_0 + A)}{s\left((s+\sigma)^2 + \omega_d^2\right)} = \frac{A}{s} + \frac{Bs + C}{(s+\sigma)^2 + \omega_d^2}$$

$$\Rightarrow \theta(s) = \theta_0 + A\left[\frac{1}{s} - \frac{(s+\sigma)}{(s+\sigma)^2 + \omega_d^2} - \frac{\sigma}{(s+\sigma)^2 + \omega_d^2}\right] \Rightarrow \theta(t) = \theta_0 + A\left(1 - \frac{\omega_n}{\omega_d} e^{-\sigma t} \cos(\omega_d t - \phi)\right)$$

When we change the arm angle from 0 to 60 degrees, we have observed the response of the system when Kp is 5 as shown in Figure2.

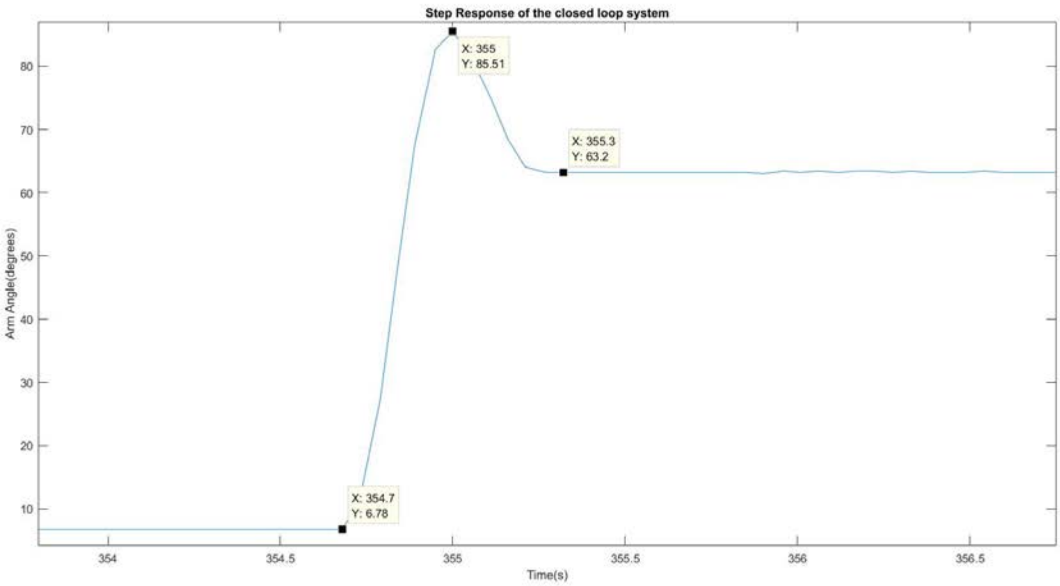

Figure 2: The response of the system to the input expressed above when Kp is 5.

From the measured values, we calculated the system parameters β and ɣ as:

β = 6.185

ɣ = 23.85

We have plotted the step response of the system with the parameters above in MATLAB, as shown in Figure 3.

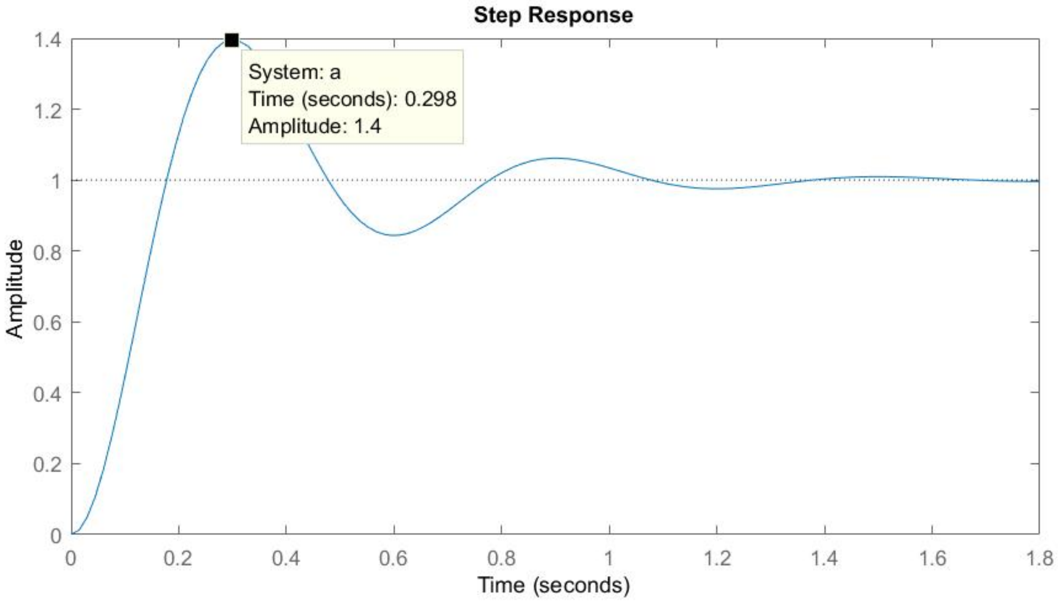

Figure 3: The theoretical step response of the system with the parameters above.

By comparing Figure 2 and Figure 3, one can see that the theoretical response is matched with the experimental result.

The system is type-1; therefore, we expect the steady state error of zero. However, in the experiment we have observed nonzero steady state error of 3 degrees. The reason for this might be the non-ideality problems, such as friction, noise of the POT, calibration errors etc. For instance, while calibrating the system, we have observed that the measured voltage values from the POT are not linearly dependent with the angle of the POT.

Step Three

a.

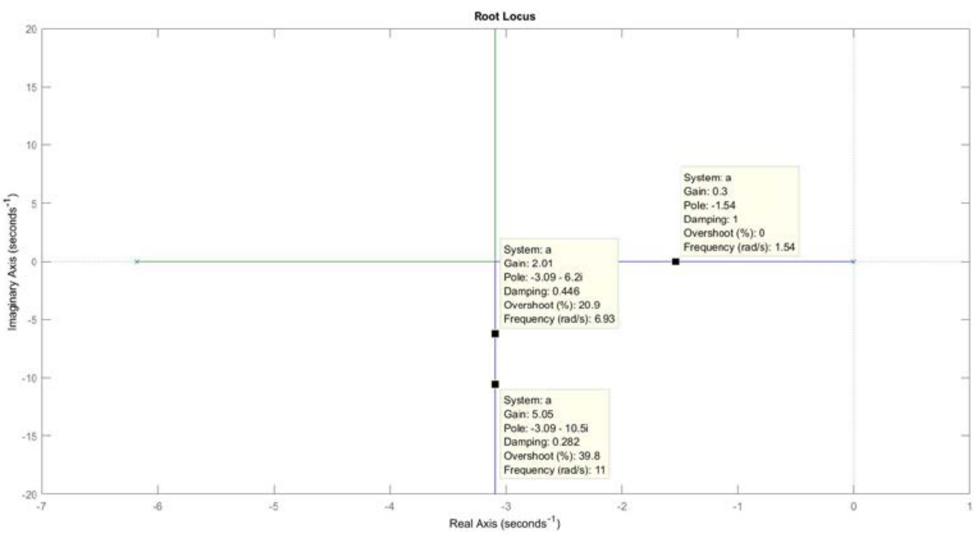

Figure 4: The root locus of the system with the parameters above.

b. We picked the K values as shown in Figure 4. There are 2 values for underdamped region and 1 for the overdamped region. We will observe the overshoot change for these values.

c. As expected, in the underdamped region when K is increased, overshoot and settling time increased, peak time decreased. For the K value corresponding to the overdamped region, we observed that the rod is not moving since the torque of the motor is not enough to make rod move.

d. In overdamped case, system never reaches steady state so we couldn’t observe the behavior of the system. In underdamped case, when the K is increased, the system responds to the flicking much faster. It is because peak time is decreased.

Step Four

We calculated the K value for the %10 overshoot system response as 1.15. However, the problem with this value is that the system cannot beat the friction force and cannot operate.

Approximate peak time for this K value is 0.72s. Settling time is 1.64s. The experimental result for the peak time is 0.84s and for the settling time is 1.06s. These results are shown in Figure5.

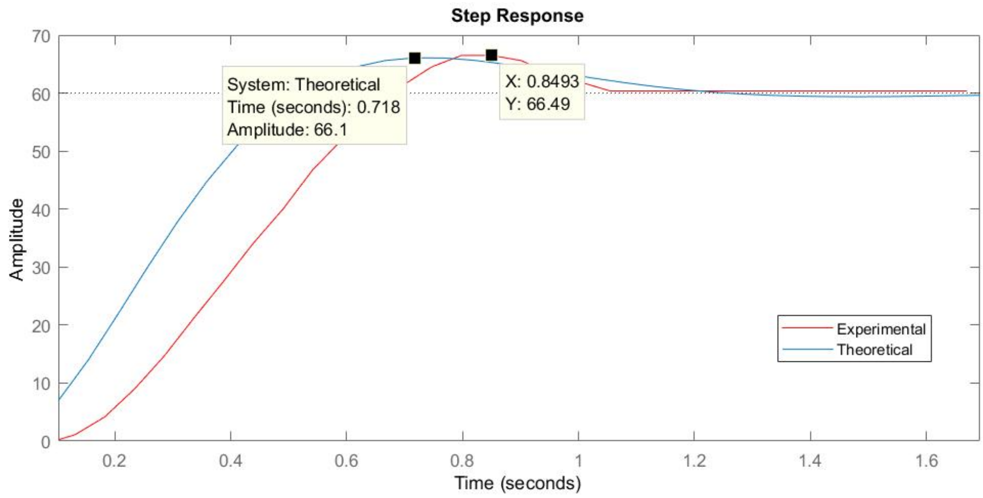

Figure 5: The experimental and the theoretical step response for %10 overshoot.

Step Five

When we set Kp and Ki as 0 and Kd to a small value, we observed that the motor tries to oppose the external applied force. Thr magnitude of the opposing force of the motor is proportional to the Kd and the moving velocity of the arm. By adding the derivative controller, we add a new pole to the system. When Kd is very large, the system goes unstable due to the added new pole, resulting in a continuous oscillation. This result is expected since the Arduino simply commands the motor to operate the opposite direction when the motor is operating in one direction and vice versa.

Step Six

We have calculated the alfa and K values for %10 overshoot and $T_4/2$ settling time as follows:

$$M_p = 0{,}1 = e^{-\frac{\pi}{\tan\phi}} \Rightarrow \ln 0{,}1 = -\frac{\pi}{\tan\phi} \Rightarrow \phi = 53{,}76 \Rightarrow \cos\phi = \xi = 0{,}59$$

$$t_s = \frac{3}{\xi\omega_n} = \frac{T_4}{2} \Rightarrow \omega_n = \frac{6}{\xi T_4} = 6{,}2$$

$$CLTF = \frac{K\gamma(s+\alpha)}{s^2+(\beta+K\gamma)s+K\gamma\alpha} \Rightarrow \begin{array}{l} 2\xi\omega_n = \beta + K\gamma \Rightarrow K = 0{,}0474 \\ \omega_n^2 = K\gamma\alpha \Rightarrow \alpha = 34 \end{array}$$

After calculating these values, we have obtained the root locus of the system on MATLAB as shown in Figure 6.

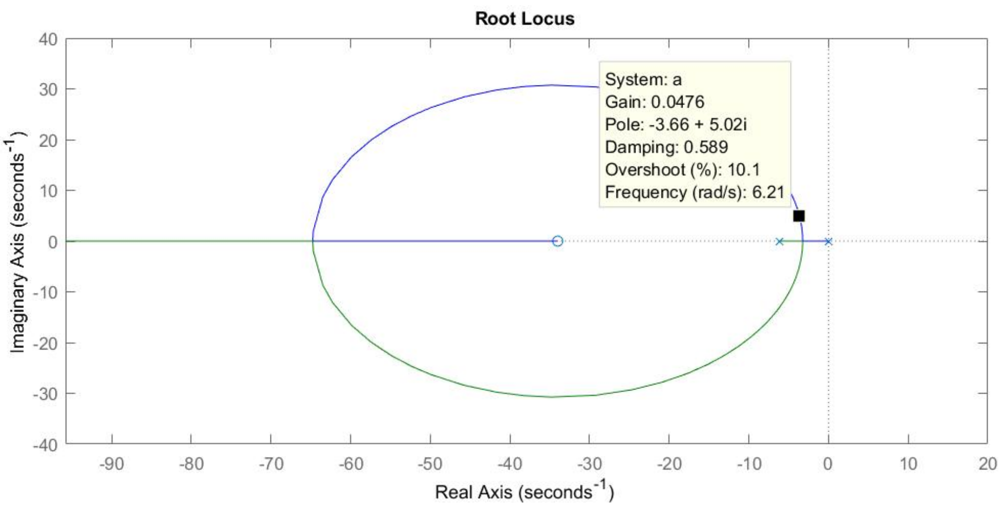

Figure 6: The root locus of the system with a PD controller having %10 overshoot.

Step Seven

For fixed Kp and Kd values, we observed that the steady state error decreases as K_I increases. However, if we increase K_I too much, system becomes unstable. The advantage of using K_I is to decrease steady state error. This is because PID controller adds 2 zeros and 1 pole to the open loop transfer function of the system. For Kp = 7, Kd = 0.9 ; we observed system response parameters as follows

	K_I=0	K_I=5
t_s	0.8s	0.7s
Mp	%60	%73
e_{ss}	4°	1°

Table 1: System response parameters according to the given parameters above.

Step Eight

For Ziegler-Nichols method, we used the following equations.

Control Algorithm	K_p	K_i	K_d
P control	$0.5K_u$	–	–
PI control	$0.45K_u$	$1.2\frac{K_p}{T_u}$	–
PD control	$0.8K_u$	–	$0.125K_pT_u$
PID control	$0.6K_u$	$2\frac{K_p}{T_u}$	$0.125K_pT_u$

Table 2: The method to obtain the PID parameters according to Ziegler-Nichols method.

a) We set Kd and K_I to zero and gradually increase Kp until we observed the constant oscillations at the output for a critical Kp value; K_U =9 with the period T_U = 0.39 s.

b)

	Kp	K_I	Kd
PD	7.2	-	0.351
PID	5.4	27.7	0.263

Table 3: PID parameters.

c) It is very difficult to tune the PID controller parameters in a complex system. By using the Ziegler-Nichols method, we found these parameters as shown in Table 3. Target performance parameters are fastest response with minimum overshoot. We have met these requirements with PD and PID controllers as shown in Figure 7 and Figure 8. The PID controller one is better than the PD controller one.

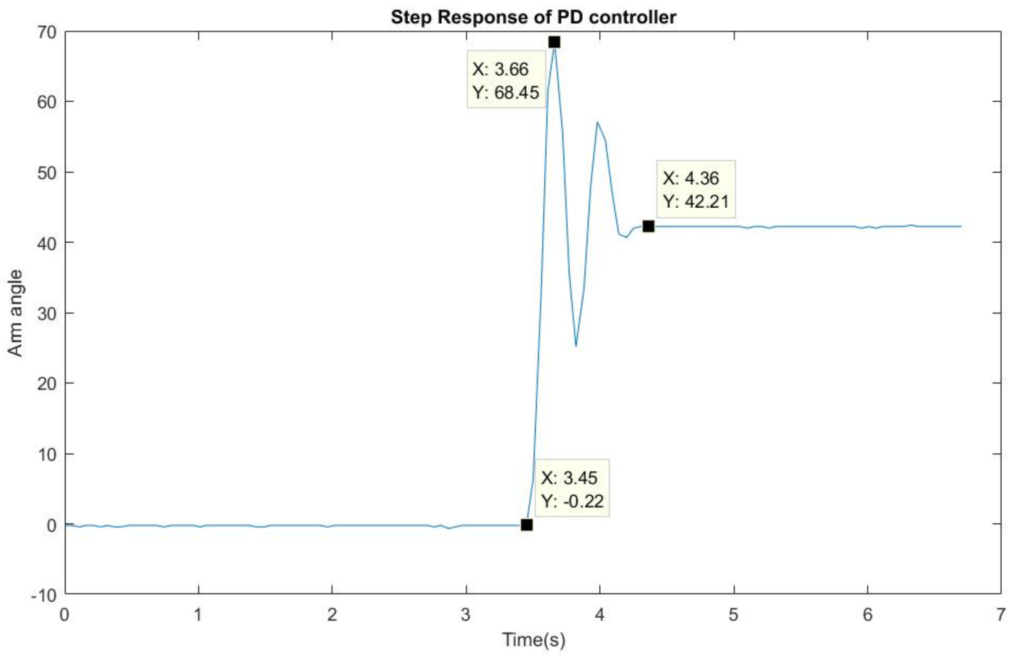

Figure 7: The step response of the system with PD controller.

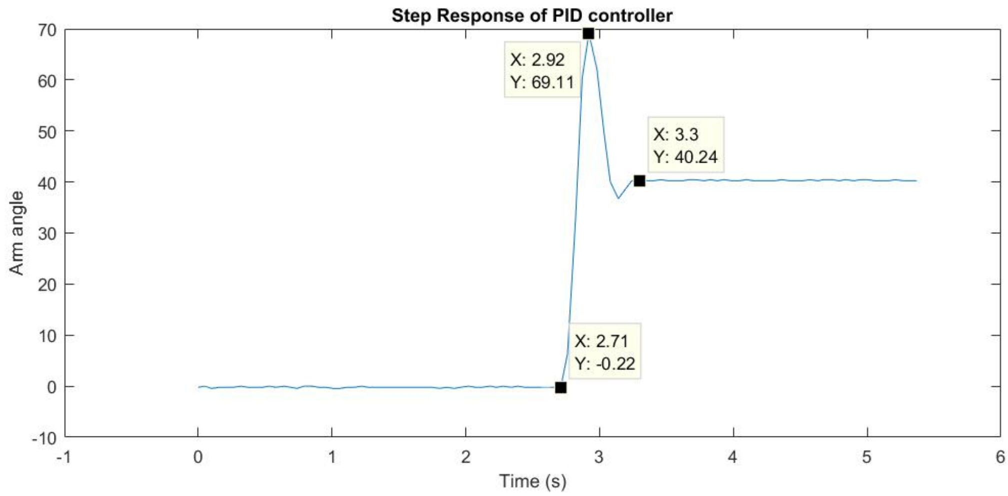

Figure 8: The step response of the system with PID controller.

Step Nine

Kp = 5 Ki = 30 Kd = 0,8. We found these values by trail- error method.

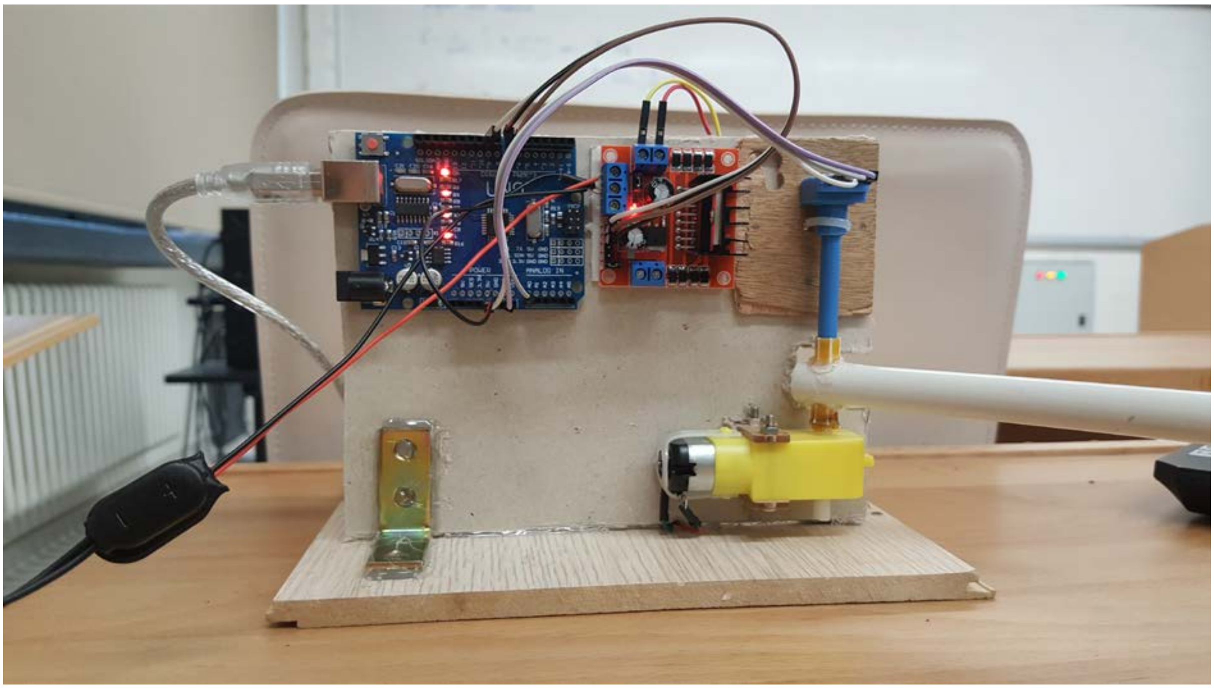

Figure 9: The real time implementation of the setup

First Video Link: https://youtu.be/jFdJ_-yZ0w0

Second Video Link: https://youtu.be/LbzB5pU2qY8