MIDDLE EAST TECHNICAL UNIVERSITY

ELECTRICAL AND ELECTRONICS ENGINEERING

EE302 FEEDBACK SYSTEMS

BONUS PROJECT

FINAL REPORT

COPE WITH COPTER

Prepared By

Merve AZMAN 2093391

Asya BAL 2093417

Rafet KAVAK 2166783

Submission Date: 17/05/2019

Table of Contents

1. Introduction

In our project, we designed one-axis helicopter by using microcontroller in order to observe how a simple control system works and can be implemented in an affordable way.

In the first part of our project, we obtained the numerical values of the unknown parameters such as time constant β_m, gain γ_m for actuator and γ, β and η for the copter arm.

For the second part, we designed a controller for our system to control it at the certain position.

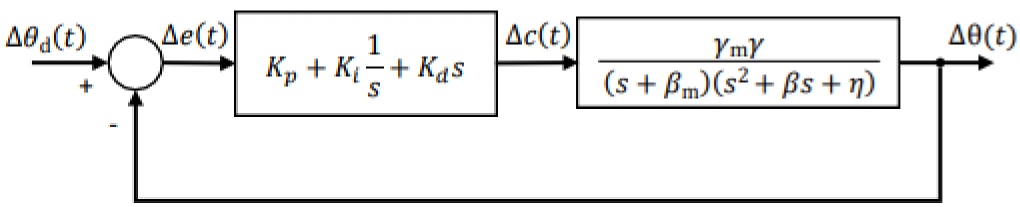

Figure 1: Block diagram of the overall system with PID controller.

As can be seen from the block diagram, c(t) depends on the error signal e(t) and the controller parameters. Therefore, by adjusting these parameters, we can stabilize our circuit.

Firstly, we found parameters of the controller theoretically and then, after we performed the steps, we compared our results and we saw the similarities and differences. These parameters decreased our error signal and oscillations of the plant when they adjusted suitably.

Then, we ended our project by finding actual upper limits for the controller parameters.

2. Experimental Results

1)

From the previous part, we found that β_m=5.56 s^{-1} and γ_m=3.55 for the propeller. Also, we found γ=2.888(after rad degree conversion), β=1.618 and η=1.844. Then, by using these numerical values, we obtained the root locus diagram of our closed loop system.

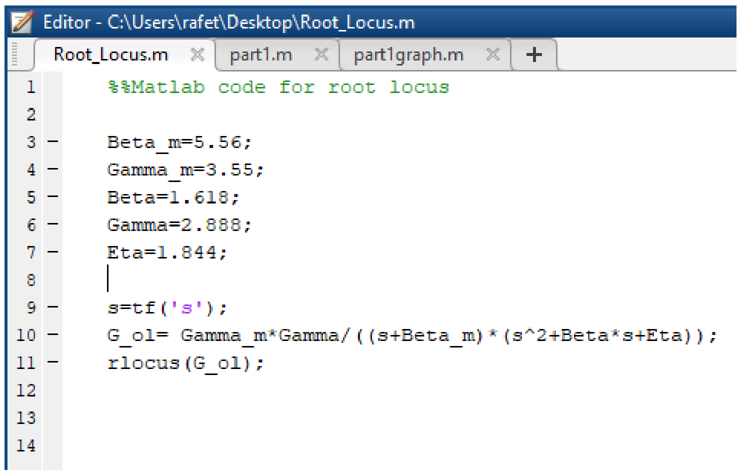

Figure 2: MATLAB code of root locus graph.

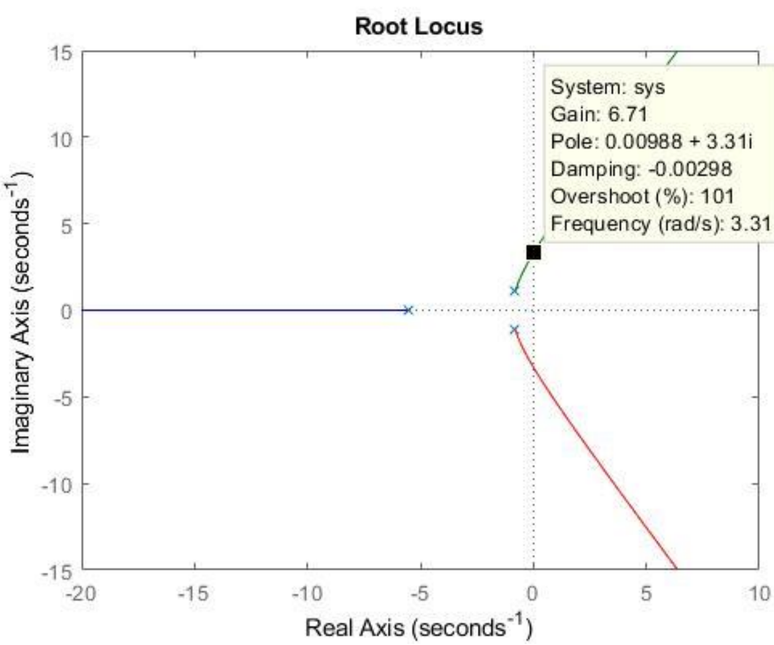

Figure 3: Root Locus of the closed loop system.

Our open loop system is stable until the Kp value becomes 6.71 theoretically.

2)

From the plot, maximum value of K_p is 6.71. At this point, our system is critically stable. Let us calculate analytically too.

$$KG(s) = \frac{K_P \gamma_m \gamma}{(s+\beta_m)(s^2+\beta s+\eta)} \tag{1}$$

Then d(s) is

$$d(s) = (s+\beta_m)(s^2+\beta s+\eta) + K_P \gamma_m \gamma \tag{2}$$

After rearranging the terms

$$d(s) = s^3 + (\beta+\beta_m)s^2 + (\beta\beta_m+\eta)s + \beta_m\eta + K_P\gamma_m\gamma \tag{3}$$

At this point, we can use Routh-Hurwitz Method for calculating jw-axis intersections.

s^3	1	$\beta\beta_m+\eta$
s^2	$\beta+\beta_m$	$\beta_m\eta + K_P\gamma_m\gamma$
s^1	$[(\beta\beta_m+\eta)(\beta+\beta_m)$-$(\beta_m\eta+K_P\gamma_m\gamma)\ ]/\beta+\beta_m$	0
s^0	$\beta_m\eta + K_P\gamma_m\gamma$	0

From the s row,

$$\beta\beta_m + \eta - \frac{\beta_m\eta + K_P\gamma_m\gamma}{\beta+\beta_m} = 0 \tag{4}$$

Then,

$$K_P = \frac{\beta^2\beta_m + \eta\beta + \beta\beta_m^2}{\gamma_m\gamma} \tag{5}$$

After inserting the values,

$$K_P = 6.59$$

We could not measure the exact intersection point on the jw-axis (0.00988), therefore we found the Kp max 6.71. If we could measure, it will become 6.59. Also, oscillation frequency at this point is 3.31rad/sec.

3)

In this part, we brought the copter arm to -45° and set the proportional gain to $K_{p,max}/4=1.67$. After that, we applied disturbances to the system by poking it with our hand, and the system came back to its equilibrium point as expected because the arm system was stable. Then, we increased the Kp value until we reach an unstable point which was Kp=2.5. However, we expected the maximum Kp value to be 6.71, so there was a difference between our theoretical finding and practical observation.

We found the frequency of oscillations as 0.9 s^{-1} as in the Figure 4. It was also a little bit different than the theoretical finding which was 3.31 rad/sec = 0.527 s^{-1}.

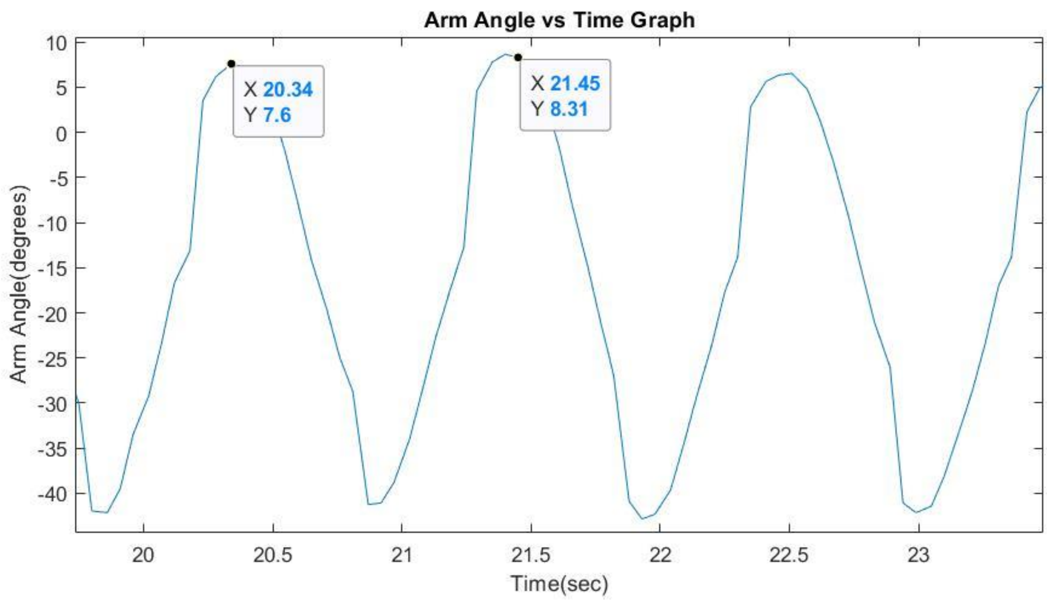

Figure 4: Arm Angle vs Time Graph.

4)

In that part, K_p value was set to 1.25 that is half of the maximum experimental value of K_p.

The system can follow the set point changes quickly. However, when we increased desired value too much, it cannot follow set point changes as quickly as before. The reason is that we linearized the system at -45° with a specific direct value and when we change it, linearity of the system also changes.

For all observations, there is a steady-state error which is an expected result since the system is run by a proportional controller. Proportional controllers have a constant steady-state error for Type-0 systems and we have modelled our system as a Type-0 system. In order to eliminate steady-state error, we need to add integral part to the controller.

The system responds disturbances. When the balance point of the system is broken by an external effect, it tries to turn back its previous position and it does.

When we adjust and stabilize the propeller arm to desired angle position by hand, the motor stops as expected.

5)

In some cases, we managed to control our arm at 45 degree when we adjusted K_p and desired value. However, after few seconds, our system became behaving nonlinear and it deviated from its position.

6)

We set Kp=Ki=0 and set Kd=0.1 initially. We quickly rotated arm angle by hand and it came back to its equilibrium point very quickly. At the same time, motor speed followed an unperfect degraded sinusoidal wave. From this, we can conclude that with derivative controller the system responses error changes quickly. We know from our background information that Kd is used to suppress overshoot and oscillations and this is consistent with our experimental finding.

When we increased Kd further, we observed that after some value, the arm starts jittering. In our experimental setup, $K_{d,max}$ where the jittering starts is found as 0.5. Motor command is saturated when Kd>0.5 and it jumps between its maximum and minimum values like a square wave with disturbances.

7)

When we performed this step, firstly, we adjusted K_p and K_i to zero. After we increased the K_d, we realized that motor speed is increased when we poke the arm i.e., when we change the angle, the system tried to reposition itself.

When we increased K_d after 2, the system began oscillating between its minimum and maximum values. The system jittered also when K_d is less than 2 but not as much as when $K_d \geq 2$. After selected K_d=0.25, we plotted the root locus diagram.

Editor - C:\Users\rafet\Desktop\Root_Locus.m

Root_Locus.m | part1.m | part1graph.m

```
%%Matlab code for root locus

Beta_m=5.56;
Gamma_m=3.55;
Beta=1.618;
Gamma=2.888;
Eta=1.844;

s=tf('s');
G_ol= Gamma_m*Gamma/(s^3+(Beta+Beta_m)*s^2+(Eta+Beta*Beta_m+0.25*Gamma*Gamma_m)*s+Beta_m*Eta);
rlocus(G_ol);
```

Figure 5: MATLAB code of root locus graph.

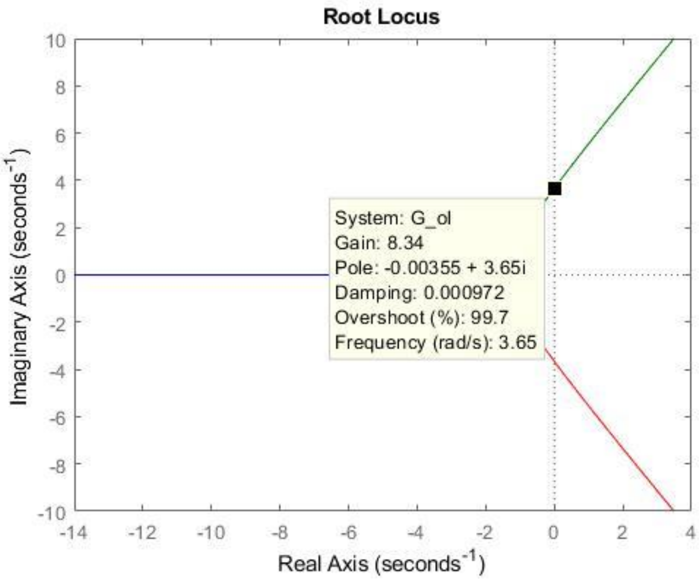

Figure 6: New Root Locus Diagram for Kp.

8)

In this case, new maximum value $K'_{p,max}$ becomes 8.34.

9)

We set K_d=0.25 and K_p= $K'_{p,max}$/4=2.1 and changed the desired value. The system responded to the change as expected since it is stable. Then, we applied external disturbances and observed a steady state error, again we observed similar steady state error in the previous part because derivative controller has no effect on the steady state error.

Then, we increased K_p further and when we reached 3, it became unstable. It is again smaller than our theoretical expectation, but accepted thinking about the non-idealities of the system.

10)

With the parameter settings in part 9, the system could be controlled for positive angles. We set the set point to +45° and applied disturbances and small changes to the set point. The system came back to its equilibrium point eventually but with a relatively higher error. Also, it takes more time to reach the stable point for positive angles compared to negative ones. This means that settling time is higher in positive angles.

11)

We set Kd=0.25, Kp=2.1 and Ki values so that the system is stable. We observed that settling time and maximum overshoot of the system increased. Moreover, the steady state error decreased significantly, and it became almost zero (oscillates around zero). This is probably due to non-linearities of the system. From all these observations we can say that stability of our system increased.

These results are consistent with our knowledge on PID controllers. They are 3^{rd} order systems and have 3 variables, so arranging these variables appropriately, we can have a system having good steady state and transient response at the same time.

According to SumMax value (0.5), we adjusted the Ki term around 150. However, when we increased the Ki, our overshoot is also increased. If we want to decrease it, we need to use more effective derivative controller. Basically, this trade off will ended up according to the system specifications.

3. Part List & Workload Distribution

3.1. Part List

We constructed our project with

- A single degree of freedom propeller-lifted-arm (Helicopter-Arm Set, we used chopstick for arm)
- An Arduino Microcontroller
- L293D Motor Driver
- 5V-3A Switching Power Adapter & Connector(2.5mm) (Then we used a DC power supply)
- 5kΩ low friction potentiometer
- 2.2µF Capacitors, 0.47Ω Stone resistor, Jumpers, Breadboard.

3.2. Workload Distribution

We have done the project all together and each team member participated in each part of the project equally. After doing all the steps together and discussing the comments and calculations, we shared some parts of the project to prepare the reports and workload distribution can be found in the table below.

Merve Azman	Finding Motor Parameters(Part 1) & MATLAB & Report & Demonstration of the circuit
Asya Bal	Finding Controller Parameters(Part2) & Report & Demonstration of the circuit
Rafet Kavak	Construction of the Circuit & Software works & Report & Demonstration of the circuit

4. Conclusion

In the first part of this project, by using the general linearized model, parameters of the setup were determined, namely time constant (βm) and gain (γm), and copter arm parameters (β, η, wd and γ).

In the second part, first we examined the root locus of the system for proportional gain controller by using the parameters found in previous part. This step gave us opportunity to have a reasonable opinion about maximum proportional gain value in which the closed-loop system is critically stable. After that, the system was observed in different conditions and gain values. Concepts in feedback systems course such as steady state error, settling time, overshoot, oscillation, stability conditions are studied in detail by using the constructed system. Some properties of different kind of controllers, discussed in the lectures theoretically, approved by observing them in a real-life application, i.e., by using cheap propeller system. For example, for both P-controller and PD-controller there were a constant steady-state error. PD-controller has better transient performance than P-controller. We also observed that PID controller has almost zero steady state error.

In conclusion, it was a good, educational project about feedback systems. It gave us more insight about PID controllers.

5. References

[1] EE302-Feedback Systems-Bonus Project Part I. METU, Department of Electrical and Electronics Engineering. Retrieved from https://odtuclass.metu.edu.tr/pluginfile.php/274083/mod_resource/content/0/Bonus%20Project%20-%20Part%201.v3.pdf

[2] EE302 - Feedback Systems -Bonus Project - Part II. METU, Department of Electrical and Electronics Engineering. Retrieved from https://odtuclass.metu.edu.tr/pluginfile.php/281132/mod_resource/content/0/Bonus%20Project-Part%202.pdf

[3] Nise, N. (2011). Control systems engineering. New York, NY: John Wiley & Sons.