Middle East Technical University

Department of Electrical and Electronics Engineering

EE302-FEEDBACK SYSTEMS

SPRING 2016 TERM PROJECT

PART-II REPORT

---

12 June 2017

Group MMB

Berk İSKENDER	2031920
Asena Melisa SARICI	2031284
Mehmet Mert BEŞE	2093482

Table of Contents

Project Description

In this project, we are required to use the knowledge that we gained during the lecture of Feedback Systems. During this project, we try to model a continous time system which contains a motor, a propeller and an arm and try to control the arm of this copter such that it remains in a desired position with no regards to the external effects. When given the externals disturbances to the system, sytems returns to its stable desired position without minimum oscillations and with minimum settling time and steady state error.

In the project, we use an Arduino to control the motor, a l293d motor driver and extra components specified in the component list. Arduino is a digital distrete time controller and in the borders of Nyquist rate, we can model our continuous time system as an appropriate discrete time system. With this idea in mind, we use PWM to drive the motor with Arduino and use PWM to measure the voltage as a analog input to Arduino to detect the arm angle of the copter.

Following the specified lab sessions and using the codes provided in these sessions, we accomplished to find the desired copter parameters deriving equations for the given inputs.

Components Used

- Arduino Uno (clone) + USB cable
- L293D Motor Driver IC
- 5 K Ω potentiometer
- Jumpers (prestripped Male/Male + Female/Male)
- Breadboard (small size)
- Heliarm set (Motor- Arm- Propeller)
- 5V DC Switching Adapter (15 Watts – 3A)
- 3.3 K Ω , 1 Ω resistors ,2.2 µF capacitors

Part I
Parameter Calculation Revised

1. Corrected Actuator Parameters

In this part of the project, we realized that using the equations we provided in the Part I of the project report we wrote, we estimaed the parameters wrong. Therefore, after performing the first part once more, we calculated these parameters so that we can use them in the second part of the project as required.

In figure 1. step response is measured. direct value is changed from 98 to 220 and the speed of the propeller increased abruptly as seen. We estimated a value in between the increasing line and using the provided equations once more:

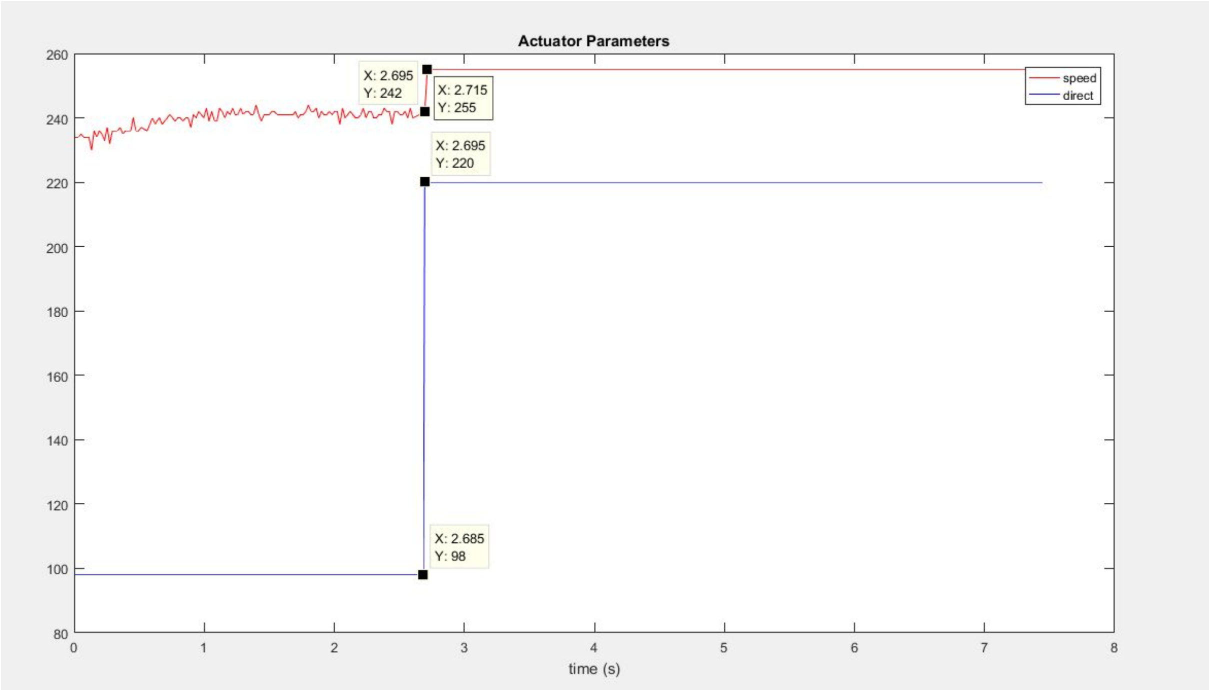

Figure 1. Step response of the Propeller

We consider V(s) as the step input with magnitude 220-98=112 in our system.

And w will be Ω(s) with

$$\frac{112}{s}\cdot\frac{\delta_m}{(s+\beta_m)}$$

Taking the inverse Laplace of this equation:

$$L^{-1}\left\{\frac{A}{s}+\frac{B}{(s+\beta_m)}\right\}$$

With A= $\frac{\delta_m}{\beta_m}$ and B=-A.

The result of this is the following equation with

$$\omega(t) = 112\cdot\frac{\delta_m}{\beta_m}\cdot(1-e^{-\beta_m\cdot t}) + 242$$

solving for some Δt we read and t=0, ∞ and by equating them to the speed values we read from the plots we obtained using MATLAB, we can find the parameters.(Table 1).

Time	Speed
t=0 sec	Approximately 242 rad/sec
Δ t=0.005sec	248.5 rad/sec
t=∞	255 rad/sec

Table.1 Speed-Time Values Read from MATLAB

From these when t=∞,

$$\frac{\delta_m}{\beta_m} = 0.11$$

And t=0.403msec,

$$0.11 \cdot (1 - e^{-\beta_m \cdot 0.005}) = 0.0542$$

Solving the equation

$$\beta_m = 135 \quad \text{and} \quad \delta_m = 15$$

2. Corrected copter arm parameters

The derivation of the arm angle is left as it is. We performed the experiment and plotting the system response, we obtained figure 2.

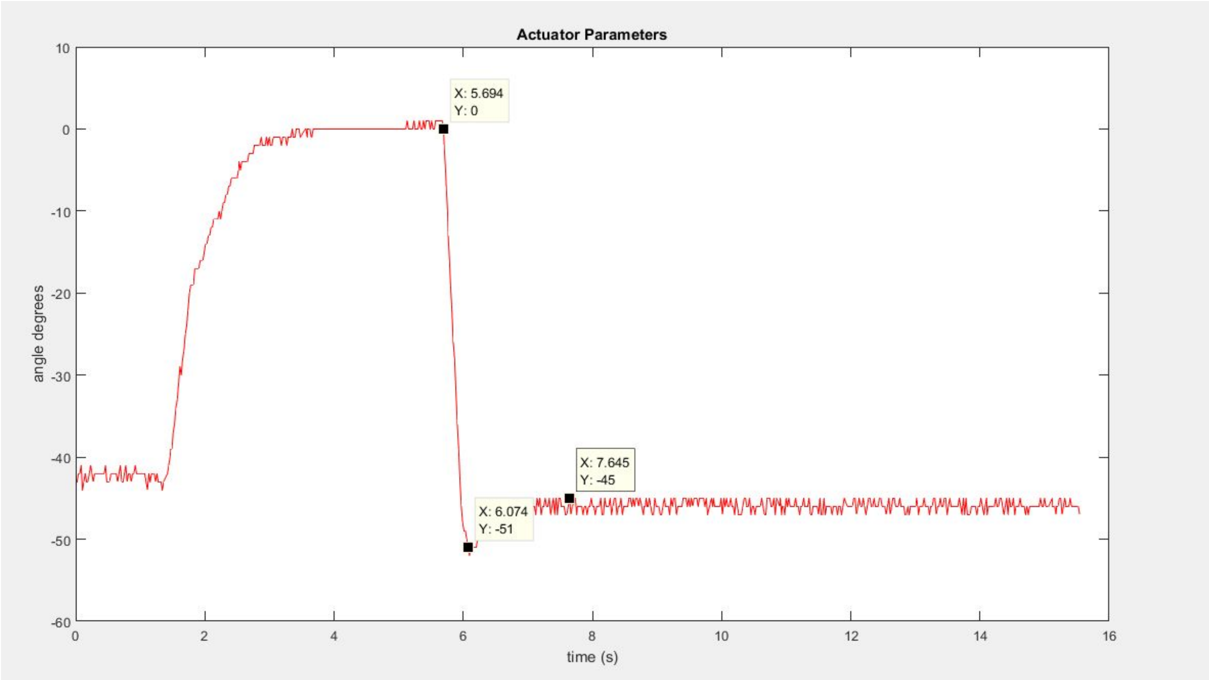

Figure 2.Results of copter arm angle experiment

Here as stated on the figure we found

$$t_p=0.38 \text{ sec}$$

and applying the givn equations in the guideline provided other parameters are found as:

$$\omega_d = 8.27$$
$$\beta = 11.578$$
$$\eta * \beta_m = 2430$$
$$\gamma = 2430/\gamma_m = 2430/15 = 162$$

Here, the derivation for the 2nd order response belonging to the first part is left as it is:

$$\frac{d^2}{dt^2}\Delta\theta + \beta\frac{d}{dt}\Delta\theta + n\Delta\theta = 0$$

$$\Delta\theta(0) = A,\ \frac{d}{dt}\Delta\theta(0) = 0$$

Also it is given that the system is underdamped, β <4n.
Since the system is 2nd order, it is possible to start with the general 2nd order step response,

$$\Delta\theta(s) = \frac{\omega_n^2}{s(s^2+\beta s+n)} = \frac{K}{s} + \frac{Ls+M}{s^2+\beta s+n}\ (1)$$

where K=1, then multiply 1 by s and take s=0.

$$\Delta\theta(s) = \frac{1}{s} + \frac{Ls+M}{s^2+2\zeta\omega_n+\omega_n^2}$$

$$\omega_n^2 = (1+L)s^2 + (\beta+M)s + n,\ where\ L =- 1\ and\ C =- \beta$$

$$\Delta\theta(s) = \frac{1}{s} - \frac{s+\beta}{s^2+\beta s+n}\ where\ L\{e^{at}cosbt\} = \frac{s-a}{(s-a)^2+b^2}\ and\ L\{e^{at}sinbt\} = \frac{b}{(s-a)^2+b^2}$$

$$s^2+\beta s+n = s^2+\beta s+\frac{\beta^2}{4}-\frac{\beta^2}{4}+n = (s+\frac{\beta}{2})^2+n(1-\frac{\beta^2}{4n^2})$$

$$\Delta\theta(s) = \frac{1}{s} - \frac{s+\beta}{s^2+\beta s+n} = \frac{s+\frac{\beta}{2}}{(s+\frac{\beta}{2})^2+n^2(1-\frac{\beta^2}{4n^2})} + \frac{\frac{\beta}{2}}{n\,(1-\frac{\beta^2}{4n^2})}\frac{n\,(1-\frac{\beta^2}{4n^2})}{(s+\frac{\beta}{2})^2+n^2(1-\frac{\beta^2}{4n^2})}$$

$$\Delta\theta(t) = [1 - e^{-\beta/2t}cos(n(\sqrt{(1-\frac{\beta^2}{4n^2}})t) - \frac{\frac{\beta}{2n}}{(1-\frac{\beta^2}{4n^2})}e^{-\beta/2t}sin(n(\sqrt{(1-\frac{\beta^2}{4n^2}})t)\ ]u(t)$$

$$\omega_d = \frac{\sqrt{4n-\beta^2}}{2}\ and\ define\ angle\ \phi\ as\ \phi =- atan(\frac{\beta}{2\omega_d})$$

Therefore, the solution has the form of,

$$\Delta\theta(t) = 1 - \frac{\sqrt{n}}{\omega_d} e^{-\beta/2t} cos(\omega_d t + \varphi)$$

However, in our case the magnitude of the unit step is A instead of 1 and it is not a 0 to A response. Due to the decrease in the angle of the copter, the response is from 0 to -A. (which can be considered as A to 0 also) Doing the relevant changes in the equation(scaling the magnitude, taking the symmetric of the equation and adding A to shift the start point to A):

$$\Delta\theta(t) = \frac{A\sqrt{n}}{\omega_d} e^{-\beta/2t} cos(\omega_d t + \varphi)$$

Part II

1)

Using the system block in figure 3 and the parameters derived from the Part I Corrected, rootlocus is obtained in MATLAB using the code provided in figure 4.

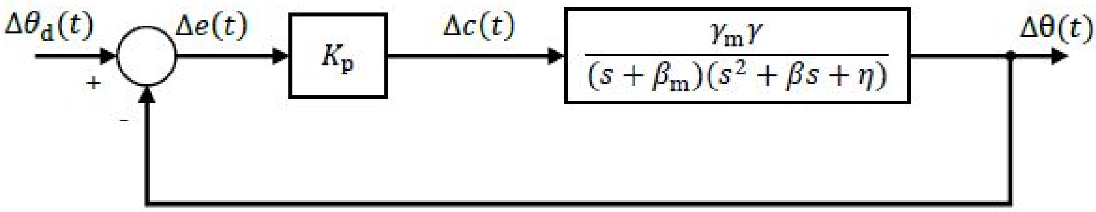

Figure 3. System Block Diagram

```
s=tf('s');
G_s=13770)/((s+135)*(s^2+11.578*s+102));
rlocus(G_s);
```

Figure 4.MATLAB code for the system transfer function and plotting the root locus

Using these, a rootlocus for our propeller sytem is obtained as shown in Figure 5. Here, the value for the proportional constant when integral and derivative constants are 0 is found as 7.4 in MATLAB. Kpmax=7.4 which is the point on the boundary of stability. We found this value from the theoretical calculations as 7.45 as well.

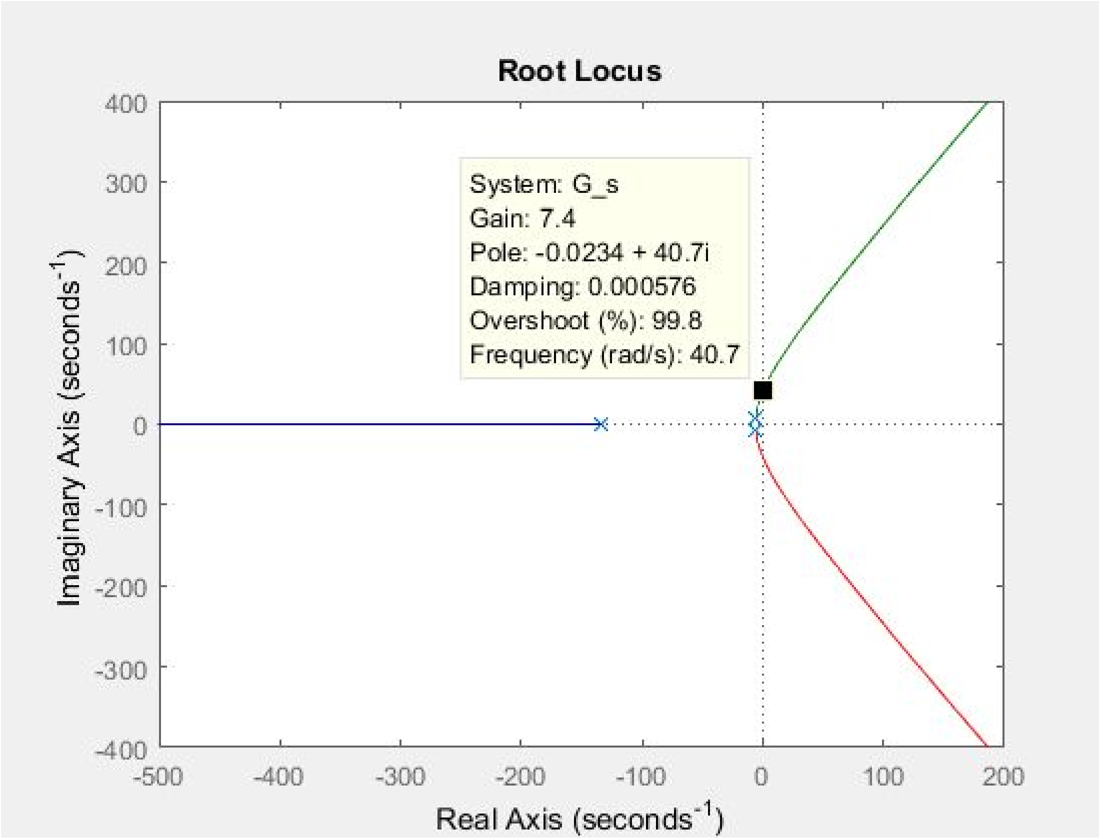

Figure 5. Root Locus for the Proportional Controller

2) $char.polynomial = (s + \beta_m)(s^2 + \beta s + \eta) + K_P s$

$s^3 + 146.578s^2 + 1665.03s + 13770 + K_P * 15 * 2060$

s^3	1	1665.03
s^2	146.578	$30900K_P + 13770$
s	$230286.76 - 30900K_P$	0
1		

$b_1 = 244056.76 - 13770 - 30900 * K_P$

$K_P = 7.45\ at\ j\omega\ axis$

$auxiliary\ polynomial\ =\ 146.578s^2 + 244056.76$

$s = \mp 40.8\,j$

$\omega = 40.8$

$f = 6.49\ Hz$ (frequency of oscillations at $j\omega$ axis)

3)

In this part, we set the Kp value to 1.85 which is one fourth of the critical one and observed that at this value, the system is stable as expected. We understood this from the fact that the copter arm went to its linearized point which is -45° after some damped oscillations. We increased the gain up until we can not get rid of the oscillations by simply waiting. The critical value for this proportional constant was found to be between 10 and 11. After this the system gets unstable. The experimental value is not the same of the theoretical one but a little bigger than that. The frequency of oscillations are observed by CSV recording as in figure 6. The frequency of oscillations in the angle are smaller than the calculated ones. The theoretical model and the actual system behave in a similar way.

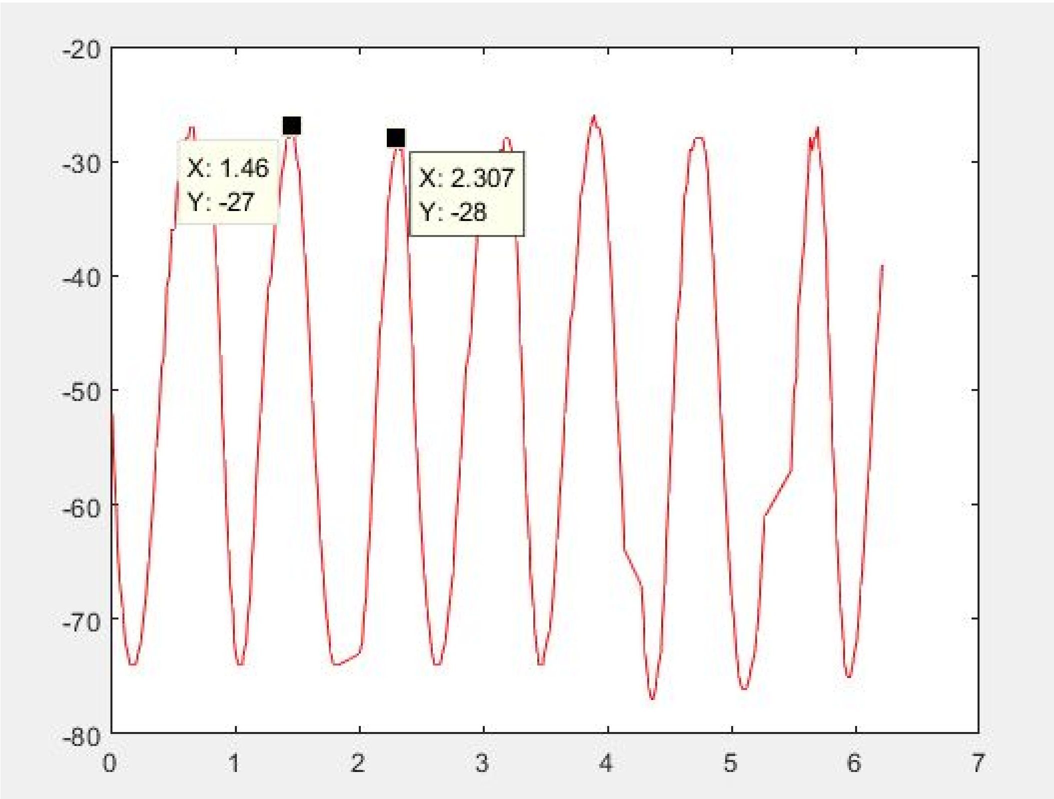

Figure 6. Oscillation when the system is in unstable region

4)

When Kp value is one half of the critical one, we changed the desired value and observed the changes in the system response. It gets harder for the system to reach the desired value when this value is increased too much.

In the steady state, the arm angle of the propeller can not reach the exact desired value hence we observe a slight steady state error at this point. This is also expected, since proportional control can not eliminate the steady state error completely but decrease it as we have observed from the graph.

At higher angles closer to 0, the damped oscillations before stabilization increased also the time required for the propeller arm to stabilize at this desired value increased whereas at lower angles closer to -90° these oscillations were damped more quickly.

When the proportional constant is set and we decrease the direct value the propeller arm angle nearly retains its value. The direct value change does not contribute to settling time considerably.

5)

For this part, we first set the desired value and arranged the direct value and derivative constants such that the arm comes to +45°. However it is hard for the system to stay at this point. Because system can not sustain its stability here when a slight disturbance occurs compared with the same amount of disturbance applied to -45°. This is because of the stable and unstable equilibrium points. Here we have two equilibrium states. As we try to increase the desired value to +45°, system started to oscillate severely at +25°, +30°. Also we observed that as we decrease Kp and increase the desired value it is easy to keep the system stable at positive values which linearizes the system at a larger angle. The reason behind the difficulty is that the system gets closer to the unstable equilibrium point as we increase the angle and it becomes harder to keep it stable. This unstability occurs due to the gravitational force's contribution to the torque and the fact that there is no negative direct value. Hence once the system passes this stable point it is not possible for it to return back.

6)

We set the Kd value to 2. At this point we experimented two different cases, one of which we applied the direct term and the other we did not apply the direct term. At the first case the propeller continues to turn (this was not the case for only-Kp-control method, the propeller velocity had decreased.) Propeller speed didn't change considerably when we increased Kd, but it slightly decreased.

When we tried this without direct value, it was easier for us to observe the propeller speed. When we quickly increased the angle by hand , as feedback effect of Kd depends on the speed of change of discrepancy between the desired value and the measured value of the arm angle, motor speed decreases faster than a slow increase, but when the measured angle is stable, no matter what the value of it is, changes in Kd didn't affect the operation and propeller speed decreases until stopping. When Kd increases MotorCmd saturated in its maximum value and we observed jittering , but when we decreased Kd Motor couldn't reached saturation values and it didn't cause jittering.

When Kd value is increased the Motor Command started to jiggle and saturate with long time instances and the motor started to slog.

7)

When Kd value is increased too much, this saturation intervals increased. We call them jitterings.This is not desired as it puts a lot of stress on the actuator and also the saturation makes the system highly nonlinear. Therefore, we decreased the value to 0.35 where we stopped observing these oscillations most of the time.

New root locus plot of the system with respect to different Kp values by taking Kd value as 0.35 is obtained by doing required changes on the characteristic equation and obtaining the equation as in proportional controller form (since we have a constant Kd value) to be able to use the rlocus function of the MATLAB. New open loop transfer function has the form:

$$G(s) = \frac{(Kp*\gamma m*\gamma)}{(s+\beta m)(s^2+\beta s+\eta)+Kd*s*\gamma*\gamma m}$$

Using the parameter values that we obtained beforehand, the root locus plot of the transfer function is obtained as:

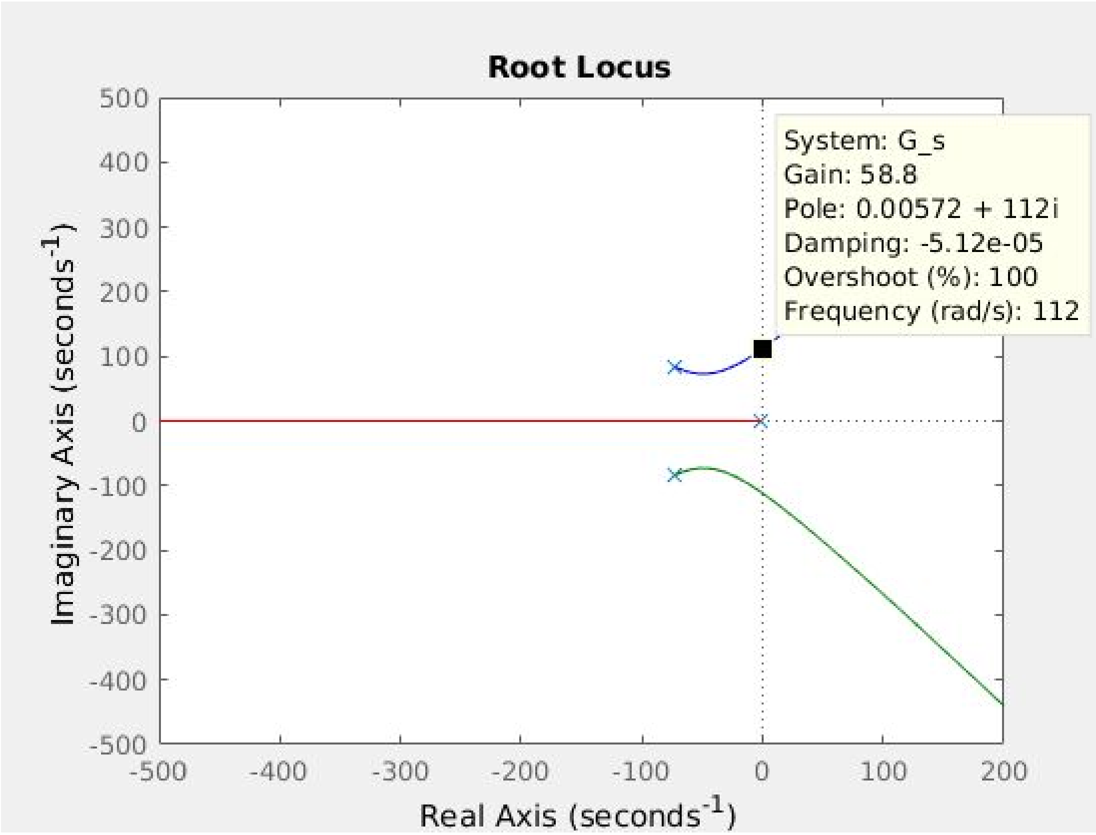

Figure 7. Root Locus with respect to Kp

introducing a derivative controller made the system more stable for different Kp values.

8) With respect to changing Kp values, open loop system became quite more stable compared to merely using proportional constant. Kp'max is almost 60 at jw-crossings. (58.8)

9)

Since our expected unstable Kp'max value is found 60, we applied 20 with 0.35 Kd value. At this value to observe whether the oscillations can be damped, we applied a small external disturbance and observed as expected that the system is still stable. Steady state arror for the linearized and stable system first oscillates and comes to balance after some time.

At this operating point when we change the desired value to positive or negative angles, the sytem can not reach that position and steady state error increases.

We can not observe the maximum Kp value since it is out of our boundary in the plots. We expect it to be close to the experimental value but smaller than that looking at the settling time we observe for Kp=20.

10)

With the derivative constant set, the system gets out of the stable region very easily at +45. It is hard to keep the arm at this position similar to only-Kp method but even harder since due to Kd the response time is smaller and a very quick change in the speed results in a very quick change in the arm angle and this as explained in part 6, causes even more oscillations and unstability compared with negative angles.

11)

Summax is set to 100 while Ki is selected 1.5. With these values we see that steady state error is decreased to 0. And system's response to any external disturbance is well balanced. This observations are plotted in figure 8.

Overall, proportional controller Kp reduces the response time and reduces but not capable of eliminating the steady-state error. The integral control Ki ,however, eliminates the steady-state error with the drawback of a probable longer response time. Lastly, the derivative contoller, Kd increases the stability of the system, decreases the overshot and makes the response time better without changing steady state error.

These effects are due to the following reasons: Kp increases the w_n term in the 2nd order closed loop transfer function (denominator) without changing the damping ratio. This means an increase in the maximum overshoot. Kd, however, increases this damping ratio. Therefore, it decreases the overshoot and due to the increase in damping ratio, the real part of the roots are increased which affect the exponential term (decaying faster) and hence minimizes both the overshoot and the response time. Derivative controller does not increase wn, hence no change in steady state error due to Kd does not occur while s goes to 0. With the addition of Ki, system transfer function's denominator becomes 3rd order and when s goes to 0, the controller constant approaches to infinity making the steady state error 0. These effects can be derived from the transfer function.

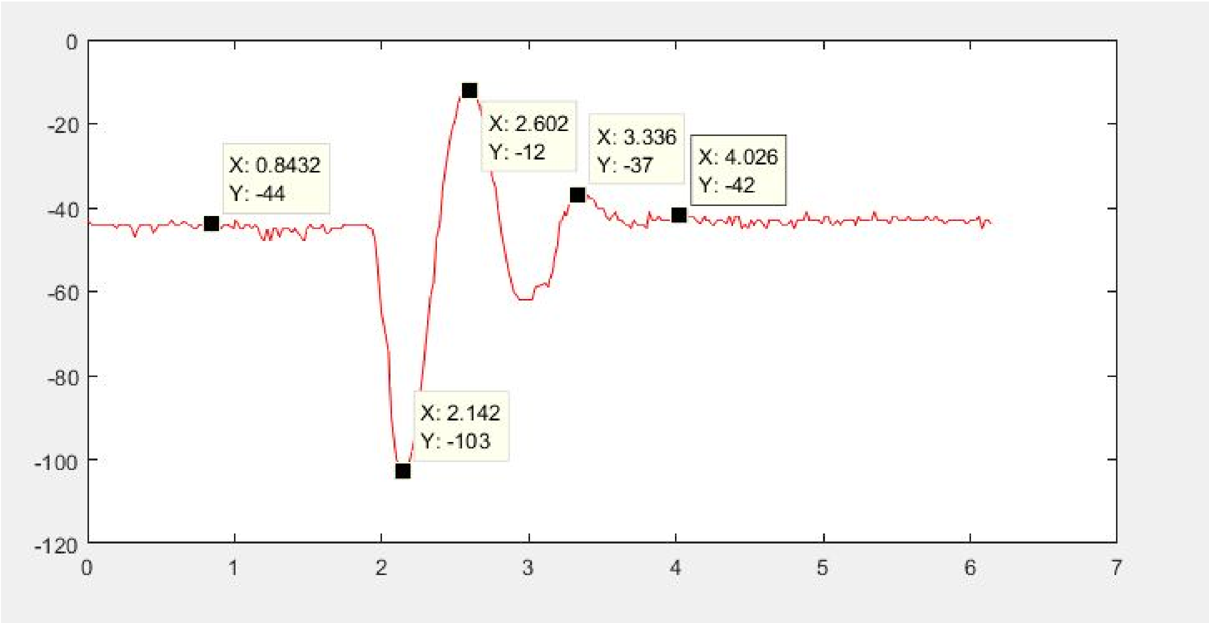

Figure 8. Response of the system to external disturbance

Discussion

In this project, we had a chance to observe and apply the theoretical knoledge we have gained through the EE302 lectures. We applied the stability rules and theorems such as Routh Hurwitz and drawing the root locus and finding the stable regions and boundaries and verified these by experimenting. We had the chance to look at the Arduino codes to increase our knowledge on microcontroller programming and we studied with the hardware implementation as well. Furthermore, it was nice to observe the changes in the response of the system when we changed the controller parameters and also to observe unstable and stable equilibrium points. Although sometimes it was hard to understand the response of the system since we have an imperfect working environment, we could not be sure about the changes we observed were exactly true and most of the times we could not get the system work in the same parameters (since there are direct control, PID control etc.) It is also notable that the POT we used lost its reference point very frequently so we needed to control and make changes on it as well. Another thing is that when the driver works a long time, it was inevitable not to get distorted results. Hence, although we were lost most of the time in these it was nice to study on this project and follow the guideline provided. We hope that we managed to fulfill the requirements during the project report and demonstration.

Project Demonstrations

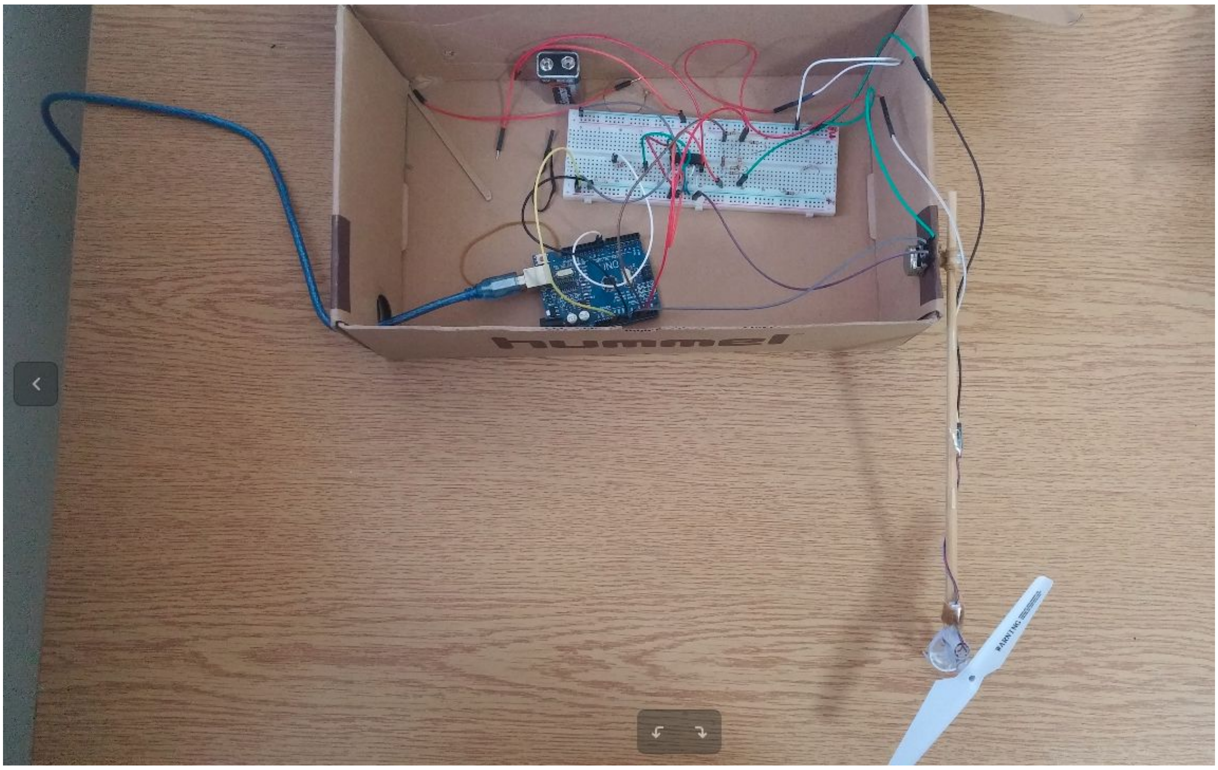

Figure 9: Top view

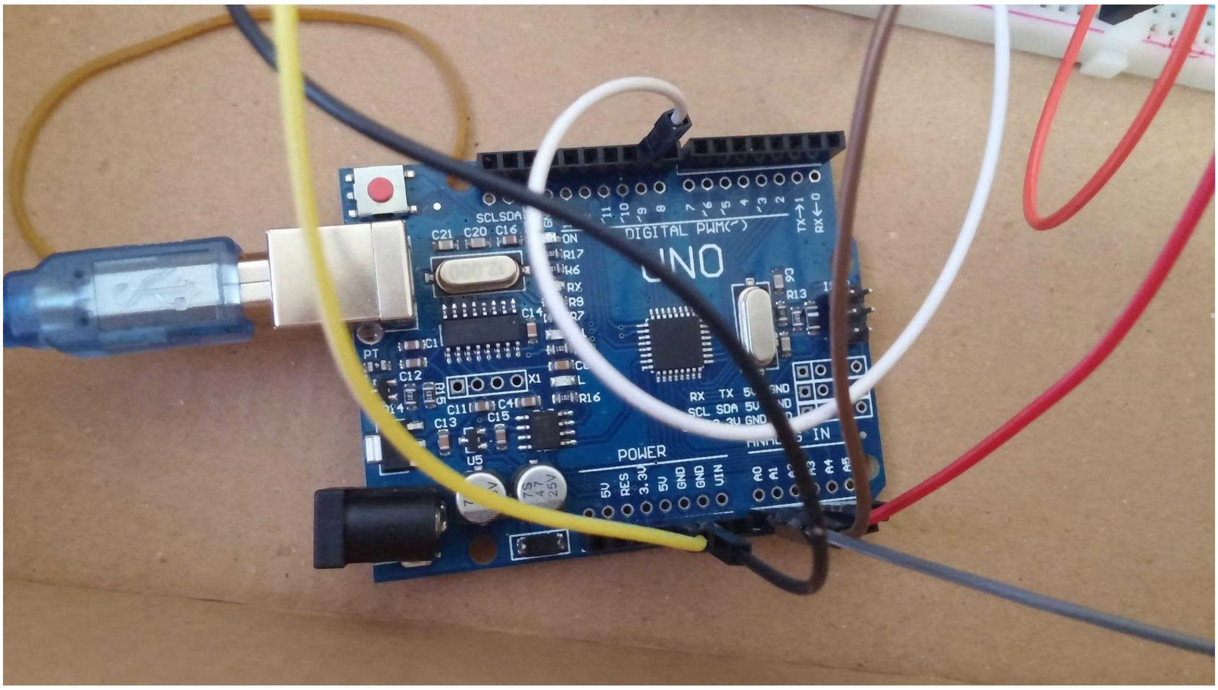

Figure 10. Arduino configuration

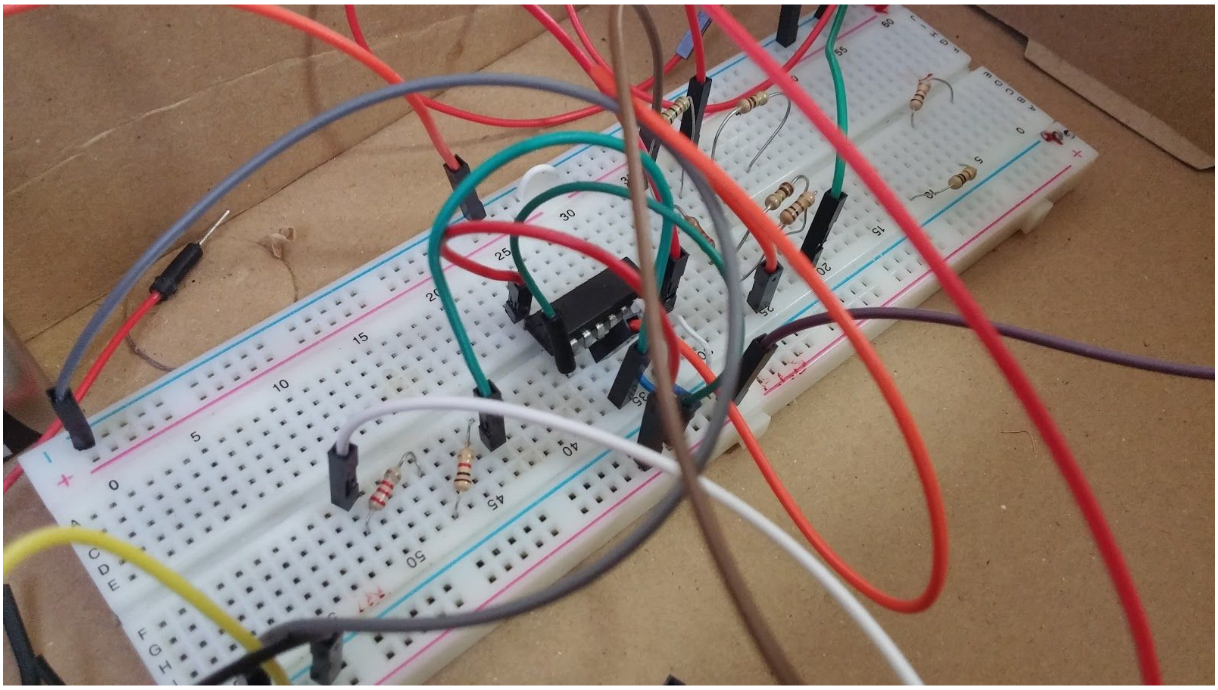

Figure 11. Driver and motor connections

.