**Koç University**

**College of Engineering**

**Electrical and Electronics Engineering**

**ELEC 304 – FEEDBACK CONTROL SYSTEMS**

**Spring 2023**

***LAB PROJECT REPORT***

Volkan Turgut

69420

08.04.2023

Contents

[Theory of The System: 2](#_Toc137531913)

[Introduction: 2](#_Toc137531914)

[Results: 4](#_Toc137531915)

[Design: 12](#_Toc137531916)

# 

# Theory of The System:

The general equation of our system was:

This is the general equation of our system and the challenging part all stems from the cos function. Due to the cos term, system becomes non-linear. In theory, after saturation, SSTE must be zero if it is a stable system and if it has integral controller in it. However, even SSTE was not perfect so it needs modification to make it almost 0. The objectives demanded from our instructor were:

* Transient performance: Given step inputs (0 to 45 and 45 to 0) in the desired angle:

• Amount of overshoot or undershoot.

• Settling time.

* Steady-state performance at a desired angle of 45 degrees.
* 0 to maximum angle step without tip over.

However, our instructor wanted us to build a good overall controller that works good at almost every operating angle.

# Introduction:

For this project, I tried to design a controller that is functioning in every possible operating angle. It served the purpose at the end of the day. However, in the process of designing the controller while trying to improve one of the three important functionalities, there were payoffs from the other functionalities. However, I tried to design the best controller overall without being aware of the weights of these functionalities and I hope I satisfied my instructor’s expectations and somewhat nailed my focus of these functionalities in an order my instructor wanted.

At first I started from dividing operating angles into parts. The designated intervals were starting point – (-30) degrees, (-30) – 30 degrees, 30 – 90 degrees. Because of the non-linear nature of our system due to gravity and circular motion of the propeller, there is a cosine term introduced in the overall system equation. Cosine has its local maxima at 0 degrees and it decreases with a very small slope around 0. I decide the interval range is between -30 and 30 because after absolute of 30, the tangent’s magnitude increases so the force acting upon the propeller decreases gradually. So, I wanted to make a clear distinction between intervals where gravity has more significance than the other interval and decide that the best angle for dividing these intervals is 30.

I first ran a matlab script for deciding on the optimum parameters for these intervals. This matlab script changes the parameters runs the simulation on software, fetches the timeseries data and computes with the given data. Later it iterates over all the parameter combinations that I choose and I looked for the best one inside these data. Even though when I tried the most optimum parameters on our physical model, every time the system became instable. However, at least I had an intuition of the ratios of the parameters and about filter parameter. Filter parameter was the most confusing part because as I tried nearly thousands of combinations on the software simulation, I found out that filter coefficient has a greater effect than the Kd parameter which was very helpful on my physical simulation.

# Results:

-90 🡪 75

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturulduTransient Time: 10.84 seconds

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturulduOver/Under Shoot: 2.77 degrees

Peak To Peak: I believe it stabilizes around 75 degrees with a little bit of fluctuations of at most 0.8 degrees after stabilizing.

0 🡪 74

Transient Time: 12.6 seconds

metin, ekran görüntüsü, multimedya yazılımı, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

A little bit stability is introduced after 72-73 degrees. It reaches the angle but it does not stabilize and keep fluctuates so that’s why transient time is not more than 12.5 seconds.

Over/Under Shoot: 75.05 degrees

While coming at 74 degrees, it overshoots with an angle of 1.05 degrees.

metin, ekran görüntüsü, multimedya yazılımı, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Peak To Peak:

After coming to the desired angle, it sees maximum of 75.4 and minimum of 72.7. So, the Peak-to-Peak value is 1.4 degrees.

0 🡪 45

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturulduTransient Time: 4.45 seconds

Overshoot: While coming to the desired angle, no overshoots occur but because of a general instability in my system, at steady state it fluctuates.

Peak To Peak: It sees 45.79 at most after reaching the desired angle therefore the maximum peak to peak can be considered as 0.8 degrees.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

45 🡪 0

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturulduTransient Time: 4.03 seconds

Over/Under Shoot: Again, I believe there is not over/under shoot while coming to 0 from 45 but it exceeds the desired angle after settling into the desired angle because of the total instability of the system which occurs at every angle at every time even after minutes of settling time.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturulduPeak To Peak:

At minimum, it comes to -0.57 degrees and at most it comes to 0.57 degrees. So, Peak to Peak is 0.57 degrees.

70 🡪 -40

metin, ekran görüntüsü, multimedya yazılımı, yazılım içeren bir resim

Açıklama otomatik olarak oluşturulduTransient Time: 7.3 seconds

Over/Under Shoot: -45.41 degrees

metin, ekran görüntüsü, multimedya yazılımı, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Peak To Peak: -40.47 to -39.33. Thus 0.67 degrees.

-40 🡪 45

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturulduTransient Time: 11.7 seconds

Over/Under Shoot: I believe no overshoot occurs but there is a peak-to-peak instability after coming to the desired angle.

Peak To Peak: 44.65 to 46.17 degrees. Thus, we can consider Peak to Peak as 1.17 degrees.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

# Design:

Inside the Matlab Simulink Model that our instructor provided, I have put a function block called OPTIMIZING PARAMETERS. I fed the PID controller block externally with outputs of my custom function block. Here is a snapshot of my function block.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Everything is carried out in this function block. As you can seei my custom function block takes three parameters as input which are desired angle (teta\_d), current angle (teta\_i) and error. It has 5 output parameters which are Kp, Kd, Ki, dmd, N. The source code of this custom function block is as follows:

function [Kp,Ki,Kd,N,dmd] = optimizeParameters(teta\_d,teta\_i,error)

dmd=0;

if(teta\_d<55 && teta\_d>30)

Kp=1.15;

Kd=0.25;

N=90;

dmd=-((16+error\*4/135));

if error>50

Ki=0.35;

elseif error>35

Ki=0.30;

elseif error>20

Ki=0.25;

else

Ki=0.15;

end

elseif(teta\_d<30 && teta\_d>-30)

Kp=1.1;

Ki=0.35;

Kd=0.3;

N=90;

dmd=6.5\*(cosd(teta\_i))+4;

elseif(teta\_d>=55)

Kp=1.05;

Ki=0.16;

Kd=0.35;

N=90;

dmd=-(abs(45-error\*8/135))-20;

elseif teta\_d<=-30

Kp=1.1;

Ki=0.7;

Kd=0.5;

N=80;

dmd=0;

else

Kp=0;

Ki=0;

Kd=0;

N=0;

dmd=0;

end

end

In this function block, I divided the intervals into 4 parts which are 30 to -30, >=55, <=-30, 30 to 55. I included another interval after 55 degrees because gravity’s effect is gradually decreasing after some point and we must be more careful so that it does not tip over. I mainly check the desired angle and decide on parameters according to the desired angle input. It first checks whether is it in the interval of 30 degrees and 55 degrees. If the desired angle is, then it enters the if statement of that interval. My parameters are:

Kp=1.15;

Kd=0.25;

N=90;

Dmd is a little bit confusing. So, with the given parameters above and let’s say Ki with 0.20, the controller behaved very aggressively. And I choose to adjust the md with a dmd value that is acting opposingly. I performed a lot of experiments, trials and come up with a function which increases -dmd when error is high and it comes to 0 as error decreases. It stabilizes the aggressive effect of Ki controller in this application process. Also, it tipped over after 30-40 degrees so in order to increase the transient performance and operating angle of this interval, I decided to use a dynamic Ki parameter for this purpose. Instead of being stabilized at 0.20, my Ki parameter started with 0.35 and decreased with error. Then I adjusted the dmd again according to this scenario and it worked well.

Later, I check whether the desired angle is between -30 and 30 degrees. My parameters are as follows:

Kp=1.1;

Ki=0.35;

Kd=0.3;

N=90;

Again, dmd is a little bit confusing because this time, I tried to adjust my dmd with the circular nature of my system. With various trials, I found the bets dmd value to be dmd=6.5\*(cosd(teta\_i))+4.

After this interval, I check whether the given desired angle is bigger than 55 or not. If it is bigger than 55, another statement is triggered and executed. In this statement the parameters are as follows:

Kp=1.05;

Ki=0.16;

Kd=0.35;

N=95;

dmd=-(abs(45-error\*8/135))-20;

Compared to other interval parameters, Ki is quite low. Therefore, the model is not fast responding to the desired angles bigger than 55. Therefore, it climbs to the desired angle with a little bit slower pace compared to other intervals. The dmd value this time is always negative because any excess md would cause the system to oscillate with instability and tip over after some point. Thus, I really kept dmd low for preventing oscillation and tip over.