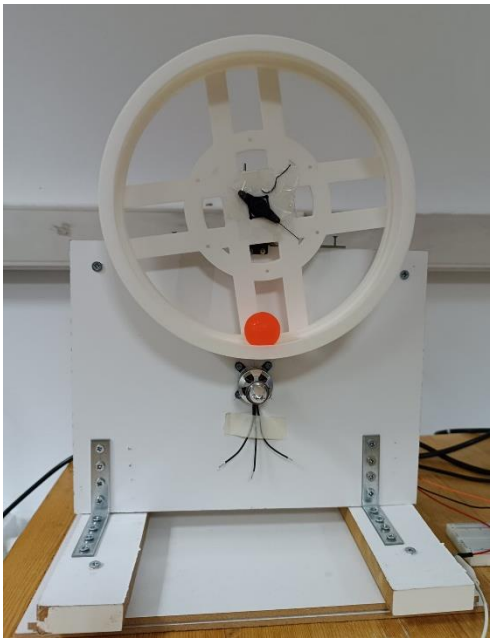
Middle East Technical University

Electrical and Electronics Engineering

EE407 - Process Control Laboratory



Experiment 7

Ball and Hoop Model

Erhan ALPAN

Muhammed Ahmet DINÇ

Muhammed Furkan BULUT

1 Objective

In this experiment, you will investigate different PID design methods such as Ziegler-Nichols (Method of Continuous Oscillations), IMC (Internal Model Control), and FOPDT (First-Order Plus Dead Time) methods on the Ball and Hoop System. The PID controller parameters will be tuned by using theoretical methods and experimental data provided for analysis. Moreover, in this experiment you will have the chance to see how these tuning methods influence system stability, transient response, and disturbance rejection. To achieve this hardware components such as Servo motor, Arduino, and webcam will be used.Ultimately, you will have a broder insight about controlling techniques of dynamic systems.

2 Experimental Setup

The ball and hoop system is composed of six parts.The each part is explained in detail as follows.

Part 1 : Orange Ball

An orange ball of radius 1.4 cm that is rolling in the hoop .The ultimate goal of the experiment is designing a PID controller in order to control the circular motion of this ball.

Figure 1 :Orange Ball

Part 2: Circular Hoop:

The circular hoop [1] which is rotated by the torque of the BLDC motor.Hoop has a diameter of 22cm and it weighs 150 grams. The orange ball is rotated by the torque caused by the friction between the hoop and the ball.

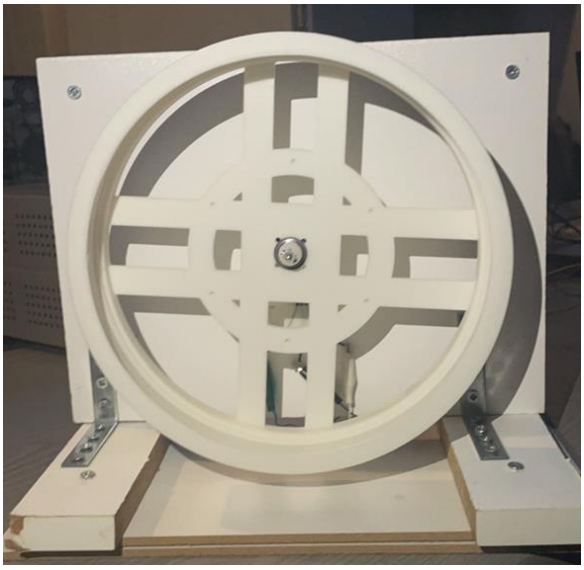


Figure 2 :Circtular Hoop (Plant)

Part 3: Servo Motor:

In this experiment, MG996R Servo Motor [2] will be used. This motor is a high torque known for its high performance and durability such that it is used in many robotic applications. It can produce a stall torque of up to 11 kg-cm at 6V. It has an operation range of 4.8 to 7.2 Volts. Another advantage of using a servo [3] motor is that it has a servo library [4] in Arduino-Microcontoller. By having this library, the operation of the motor can be easily controlled. The Arduino will send proper PWM (Pulse Width Modulation) [5] to drive the servo motor. Pulse width modulation is a very important terminology in electrical engineering that you should know by heart. Thus, we advise you to watch the video in the appendix [6] and do some research about it before coming to the experiment.Apart from that as instructors we think that every control engineer should have a deep insight about actuators and electromechanical [7] systems.We also want you to study on actuators as well.

Figure 3 : MG996R Servo Motor

Part 4:Arduino Micro-Controller:

To run the Servo Motor , the Arduino UNO R3 [8] will be used as a microcontroller. Arduino plays a critical role in the experiment in terms of supplying a PWM signal to control the servo motor. Therefore, Arduino will act as the control unit of the overall closed-loop system. It will receive the real-time angular position of the ball. This information is sent by the webcam via feedback. The information will be converted to an analog signal that Arduino can read. The data processed from WebCam will be transferred to Arduino via Serial Communication Algorithm. Serial Communication [9] [10] algorithm allows data to be sent one bit at a time between the devices using the USB. Arduino has built-in functions for serial communication as well. Thus, Arduino can interpret this data which is the state of the ball (feedback). The Arduino compares the received angular position (feedback) with the desired position (setpoint) and calculates the error. By designing the PID controller parameters, the PID algorithm in Arduino code will determine the control signal such that the board can send a proper PWM signal to the servo motor based on this PID output.

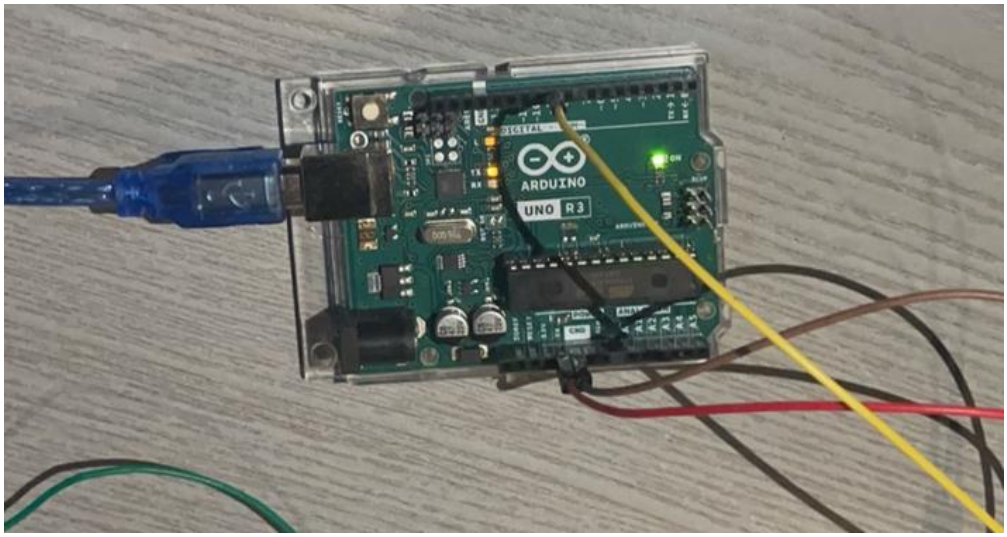


Figure 4 : Arduino Microcontroller

Part 6:The Feedback Sensor (Webcam) :

The Webcam Camera will be the feedback sensor of our system. The Webcam captures the live video of the position in real-time. With the help of the real-time image processing algorithm which is written in Python, the position coordinate of the ball can be extracted and turned into an angular position with respect to the reference point(center of the hoop). Then, this information is fed to Arduino as the feedback to calculate the error signal. The Python script for the image processing algorithm will be given in the Experimental Work Part.

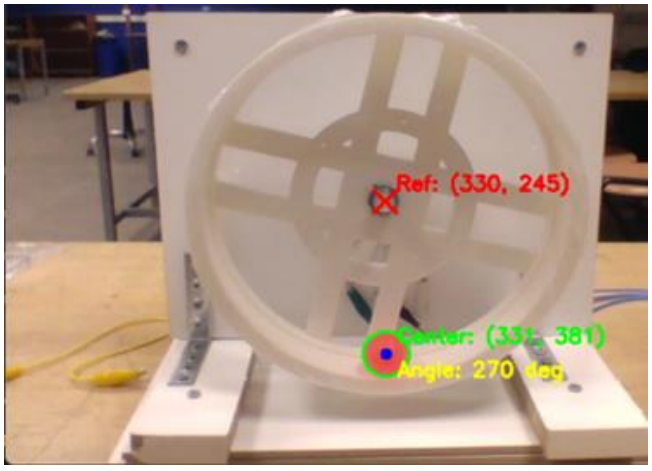


Figure 5 : The Feedback Sensor

Part 7:The Power Supply :

A simple geratech DC supply will be used in order to power the overall system.

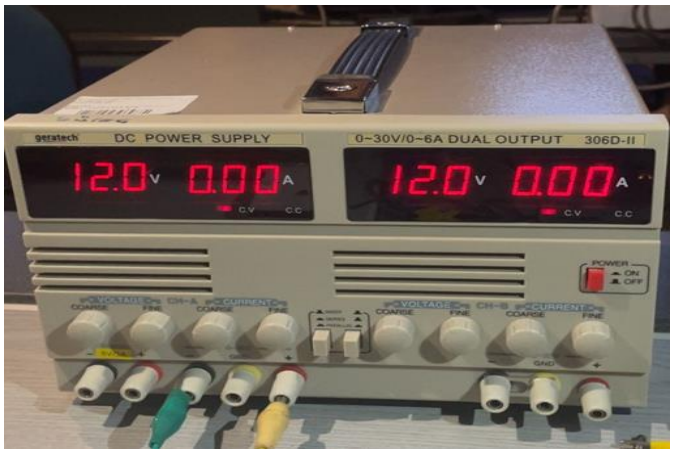


Figure 6 : The DC Supply

The Overall System:

With all of these parts the overall system looks like below.

Figure 7 : The Overall Ball ad Hoop System

Block Diagram of The Overall System:

The Block Diagram of the system is modeled in the Simulink as follows:

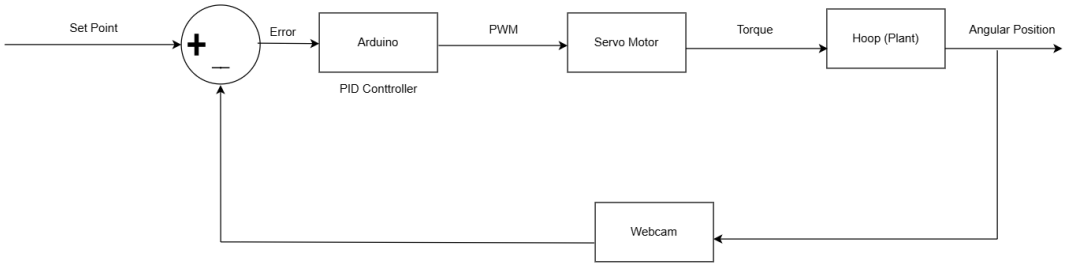


Figure 8 : The Block Diagram of System

The Wiring Diagram of The Overall System:

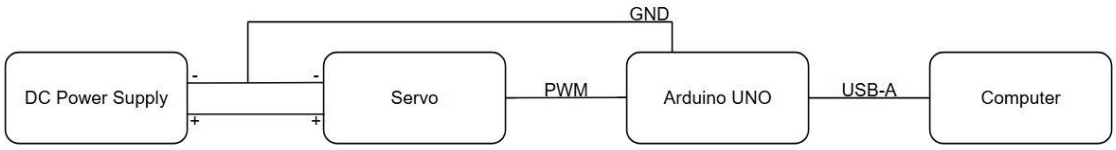


Figure 9 : The Wiring Diagram of System

3 Preliminary Work

In this Preliminary Work, you will design PID controllers by using different methods such as Ziegler-Nichols (Method of Continous Oscillations) and FOPDT(First Order Plus Dead Time).The ball and hoop system is given below.The mathematical derivations are taken from the article [11] "**Ball in double hoop: demonstration model for numerical optimal control**" written by Martin Gurtner and Jiri Zemanek .

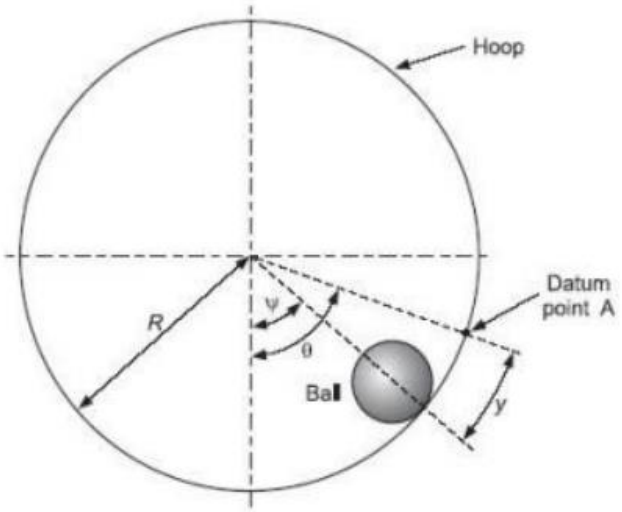


Figure 11 : The Ball and Hoop System

The analysis of this dynamical system requires Lagrangian Mechanics such that the system is governed by the Euler-Lagrange equations [12]. The key formulation of this system is the Lagrangian formulation of classical mechanics. The Lagrangian formulation is an efficient tool for analyzing complex behavior of the system. The formulation mainly uses the partial derivatives of kinetic and potential energy to find the parameter functions.
Lagrangian is defined as

$$L = T - U$$

The T stands for the total kinetic energy and U stand for the potential energy of the ball.
The Lagrangian equations of motion is

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0$$

The derivation of the characteristic coefficients come from this Lagrangian equality, and it is explained in the next section.

2.1 Background Information

System Dynamics

1. System Description and Coordinate:

The Ball and Hoop system's dynamic configuration as a ball freely rolls on the inside of a circular hoop. We will cover the dynamics of the system to understand the ball's motion in this system.
Ball motion: The ball rolls in the system where on the outer hoop's surface, which requires angular coordinates to describe its position as in Figure 12 :
- ψ: Slosh angle; an angular displacement of the ball from its equilibrium position.
- θ: Angular position of the hoop.
- y: Ball position on the hoop's surface referenced from the datum point.

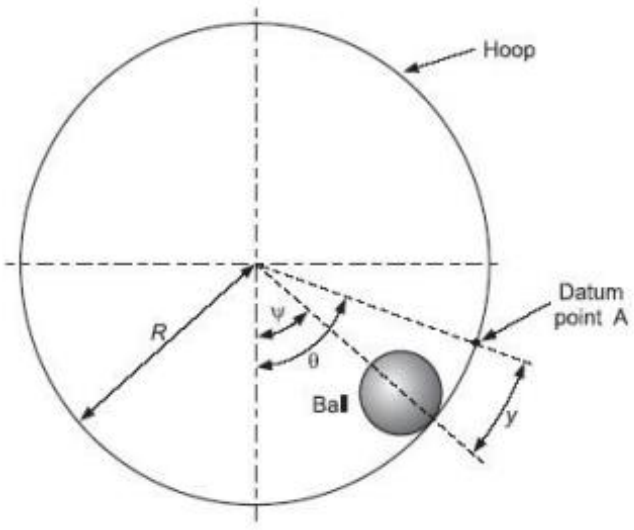


Figure 12: Spesific Angles of The Ball and Hoop System

Our assumptions on the system:
- The ball rolls with no slipping.
- The hoop is controlled instantaneously by commanding its angular acceleration (θ̈).

2. Kinetic and Potential Energy:

The system is modeled by the Euler-Lagrange equations. The energy components in the Euler-Lagrange model are defined as kinetic and potential energy.
Kinetic Energy (Translational + Rotational)
Kinetic energy of the ball as the translational motion along the hoop and the rotational motion of the ball:

$$T^* = \tfrac{1}{2}mv^2 + \tfrac{1}{2}I(\dot{\phi} + \dot{\theta})^2$$

- m: Mass of the ball.

- I: Inertia of the ball.
- v: Ball velocity.
- ϕ: Angular motion of the ball around the axis itself.

Translational and Rotational Velocity from the illustration in Figure 13:

$$v = -(Ro - Rb)\,\dot{\psi}$$
$$\dot{\phi} = (Ro / Rb)\,(\dot{\theta} - \dot{\psi})$$

- Ro: Hoop radius.
- Rb: Ball radius.

Substituting these into the kinetic energy:

$$T^* = \tfrac{1}{2}m(Ro - Rb)^2\dot{\psi}^2 + \tfrac{1}{2}I[(Ro + Rb)/Rb\,\dot{\theta} - Ro/Rb\,\dot{\psi}]^2$$

Potential Energy

Potential energy of the ball calculated from its vertical displacement on the hoop:

$$V = -mg(Ro - Rb)\cos\psi$$

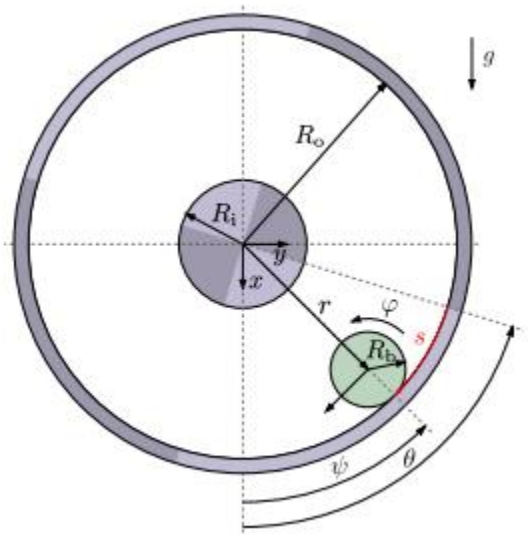


Figure 13 : Spesific Rotations and Velocities of The Ball and Hoop System

3. Friction Model:

$$D = \tfrac{1}{2}b\dot{\phi}^2 = \tfrac{1}{2}b(Ro^2/Rb^2)(\dot{\theta} - \dot{\psi})^2$$

b is a Friction coefficient in this equation.

4. Lagrange Equations and Motion:

L is the difference between kinetic and potential energy as in the equation. When we use the Lagrangian equations we can finally find the derivative equation of the motion of the ball in the

hoop.

$$L = T^* - V$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\psi}}\right) - \frac{\partial L}{\partial \psi} + \frac{\partial D}{\partial \dot{\psi}} = 0$$

Derivative equation:

$$\bar{a}o\ddot{\psi} + \bar{b}o\dot{\psi} + \bar{c}o\,sin\,\psi + \bar{d}o\theta = \bar{e}o\ddot{\theta}$$

Also, the coefficients in the equation of motion are:

$$\bar{a}o = m(Ro - Rb)^2 + I(Ro^2/Rb^2)$$
$$\bar{b}o = b(Ro^2/Rb^2)$$
$$\bar{c}o = mg(Ro - Rb)$$
$$\bar{d}o = -\bar{b}o$$
$$\bar{e}o = I(Ro/Rb)(Ro/Rb + 1)$$

5. Control Problem:

The main objective of controlling the ball and hoop system is stabilizing the ball at the bottom of the hoop and rejecting the external disturbances when this action. Therefore, first we should look at the transfer function of the system.

Deriving the closed loop transfer function of the system is out of the scope of the experiment.Therefore, the transfer function was obtained by system identification toolbox (MATLAB), and its provided below.You will use this transfer function when desigin PID controllers for the preliminary work.

Time domain experimental data is obtained by the help of image processing algorithm and webcam. The time domain response is given below.Note that it is initally at 180 degree angles with respect to reference point and it settles at its stable equlibrium which is at 270 degree angles with respect to the reference point.The response is the initial condition response such that angular position is 180 degrees initally which is illustrated below.

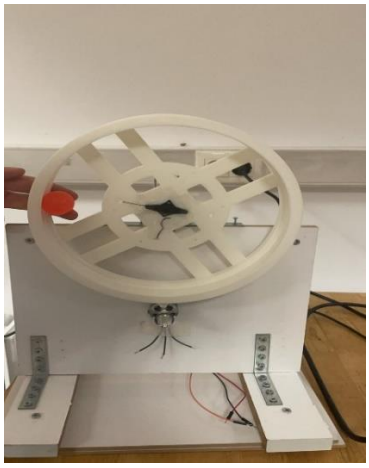


Figure 14 : The Initial Condition Set Up

There are various combinations for the number of poles and zeros that you can choose.From the data we obtianed the best combination occurs at the 5 poles and 2 zeros case.

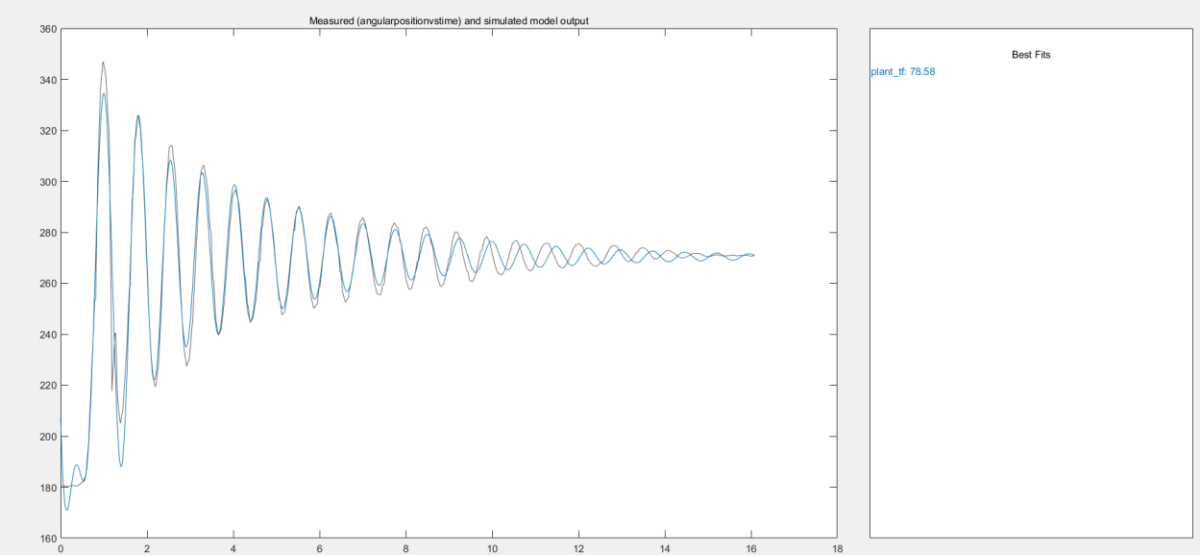


Figure 15: Best Fit Estimated Tf

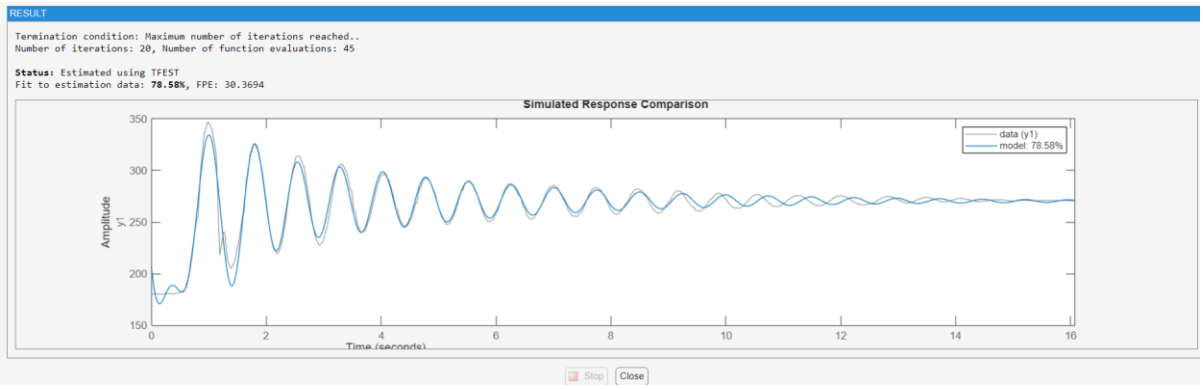


Figure 16: Estimation Results

The system identification identificates the plant transfer function as :

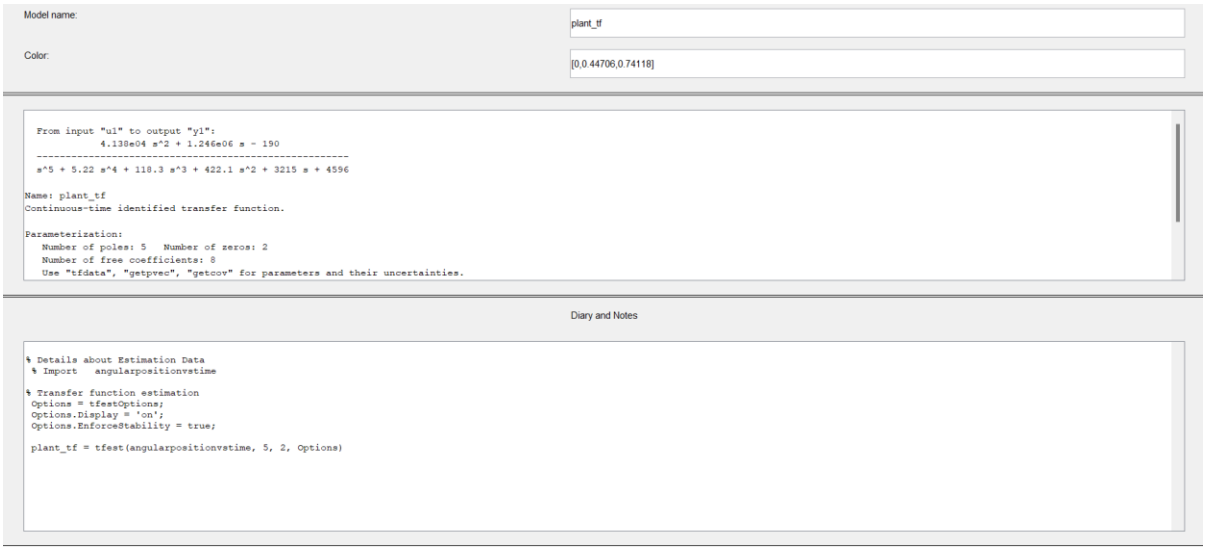


Figure 17: Estimated Tf on MATLAB

```
Status:
Estimated using TFEST on time domain data "angularpositionvstime".
Fit to estimation data: 78.58% (stability enforced)
FPE: 30.37, MSE: 29.88
```

Figure 18: MSE and Fit Percentage of Estimated Tf

The toolbox estimates the plant as a 5 poles system with 2 zeros.The fit to estimation is %78.58.

The transfer function of the hoop plant is estimated as :

```
tf_plant =

                41380 s^2 + 1.246e06 s - 190
  ------------------------------------------------------
  s^5 + 5.22 s^4 + 118.3 s^3 + 422.1 s^2 + 3215 s + 4596
```

Figure 19: Estimated Plant Tf

The time domain response of plant to initial conditon (180 degrees) (no input) is plotted below.

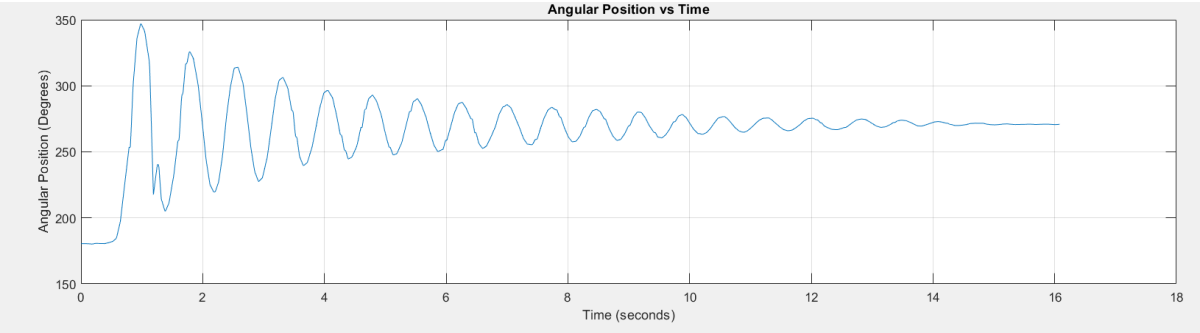


Figure 20: Time Domain Response to Initial Condition (Experimental)

The step response of the plant with 0 initial conditions is given below.

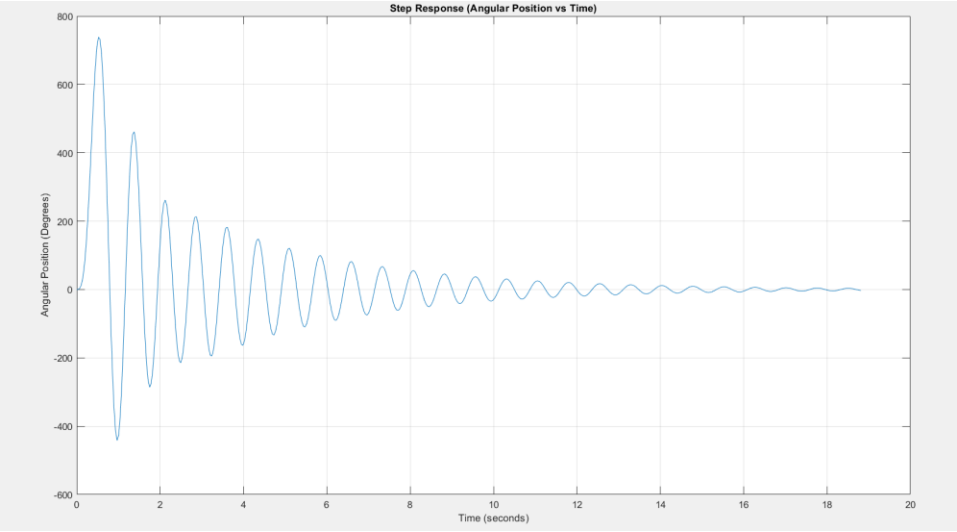


Figure 21: Step Response of the Plant

You will use this estimated plant transfer function to answer the questions in the preliminary work.

Zeros of the plant transfer function:

{-30.1113 , 0.0002}

Poles of the plant transfer function:

{-0.2607+8.429i, -0.2607-8.429i, -1.5361+6.1137i , -1.5361-6.1137i , -1.6263}

2.2 Assignments:

1. Design a PID controller using Ziegler-Nichols Method. (Hint: Since it is a 5 order tf you can use the rlocus and rlocfind commands in MATLAB to find the gain and frequency of the imaginary axis poles by using the root locus).

2. Design a PID controller using FOPDT with the help of the IMC table that is in your lecture notes.

3. Write down the advantages and disadvantages of using the Ziegler-Nichols method.

4. Write down the advantages and disadvantages of using the FOPDT method.

5. Which one is more suitable as an accurate approximation for this estimated plant model? Why?

3 Experimental Work

1. The overall Hoop system will be given to you.Check the connection between the servo motor and hoop.The Hoop should be connected tightly to the servo motor in order to avoid unstable behaviour.

2. Open and Run the Python code which has the name ***systemidentcode.py*** in the Experiment 7 folder. This code generates the experimental data in a CSV formatted file obtained from the Webcam.

3. After running this code, a camera window will pop up. Then, you will arrange the reference point (center of the hoop) by using the W, A, S, and D buttons. Make sure that the center point of the ball is measured at a 270-degree angle by the camera when the ball is at the bottom of the hoop (in its stable equilibrium point). You may need to adjust the positions of the computer and the hoop system until you observe 270 degrees in the center of the ball.

4. After setting this angle, press the R button to start getting angular position data. The advised duration is 16 seconds. Then, press the T button in order to stop recording. Close this Python code and continue with MATLAB.

5. Open the MATLAB system identification toolbox. As explained in the Preliminary Work, find the best-estimated transfer function in terms of percentage to fit data. You can change the number of zeros and poles in order to get the best-estimated transfer function.

6. Plot the root locus of the estimated transfer function in the previous step. Compare this result with the one you did in preliminary work. What is the number of zeros and poles and the fit percentage of the estimated transfer function? Are there any differences between the preliminary work and your experimental work?

7. To control the plant, you need to set the overall system according to the wiring diagram. Make sure that you make the all necessary connections.

8. Open the Arduino IDE and run the ***arduinocode*** in the Experiment 7 folder. This code provides the necessary serial communication for the next steps. Do not close the Arduino IDE until the end of the experiment.

9. Open and Run the Python code which has the name ***imageprocessing.py*** in the Experiment 7 folder. This code generates the experimental data in a CSV formatted file obtained from the Webcam.

10. After running this code, a camera window will pop up. Then, you will arrange the reference point (center of the hoop) by using the W, A, S, and D buttons. Make sure that the center point of the ball is measured at a 270-degree angle by the camera when the ball

is at the bottom of the hoop (in its stable equilibrium point). You may need to adjust the positions of the computer and the hoop system until you observe 270 degrees in the center of the ball.

11. After setting this angle, press the R button to start getting angular position data. Do not stop this code until the end of the experiment.

12. First, you need to get the approximate settling time of the plant without a controller. To do this, start the ball with an initial condition which is 180 degrees angle with respect to the reference. Measure the time until the ball returns to its stable equilibrium point. Do not forget to save this measurement.

13. You will observe the effect of the PID controller on the system. Open the Python code which has the name ***controller.py*** in the Experiment 7 folder. This code provides the PID controller algorithm to the servo motor.

14. Firstly, you will use a P controller only by setting Kp=0.5, KI=0, and KD=0 in the Python code. You will run this Python code in another terminal to be able to use both image processing and controller codes.

15. When you run this code, the hoop will rotate and set its angular position at 90 degrees and it waits 2 seconds. After 2 seconds, you will follow the same procedure in step **12** and measure the modified settling time. Stop the ***controller.py*** code.

16. Secondly, you will use a PI controller by setting Kp=0.5, KI=0.01, and KD=0 in the Python code. You will run this Python code in another terminal to be able to use both image processing and controller codes.

17. You will follow the same procedure in step **15** and measure the modified settling time. Stop the ***controller.py*** code.

18. Lastly, you will use a PID controller by setting Kp=0.5, KI=0.01, and KD=0.0005 in the Python code. Measure the modified settling time. Close all the codes that you run.

19. Make a table that includes the 4 different settling times that you have measured. Compare these results. Comment on the effects of Proportional, Integral, and Derivative actions.

20. Choose different PID gains to observe different settling times to optimize the controller design. Save these results and compare the results that you have obtained in the previous steps and comment.

Name:

Section:

Date:

4 Preliminary Work Answers

1. Note that the new open loop tf is KcGp(s) .By using the root locus script that is in code appendix, the ultimate gain and the corresponding pole locations are given as follows.

Root Locus for the Zigler-Nichols method:

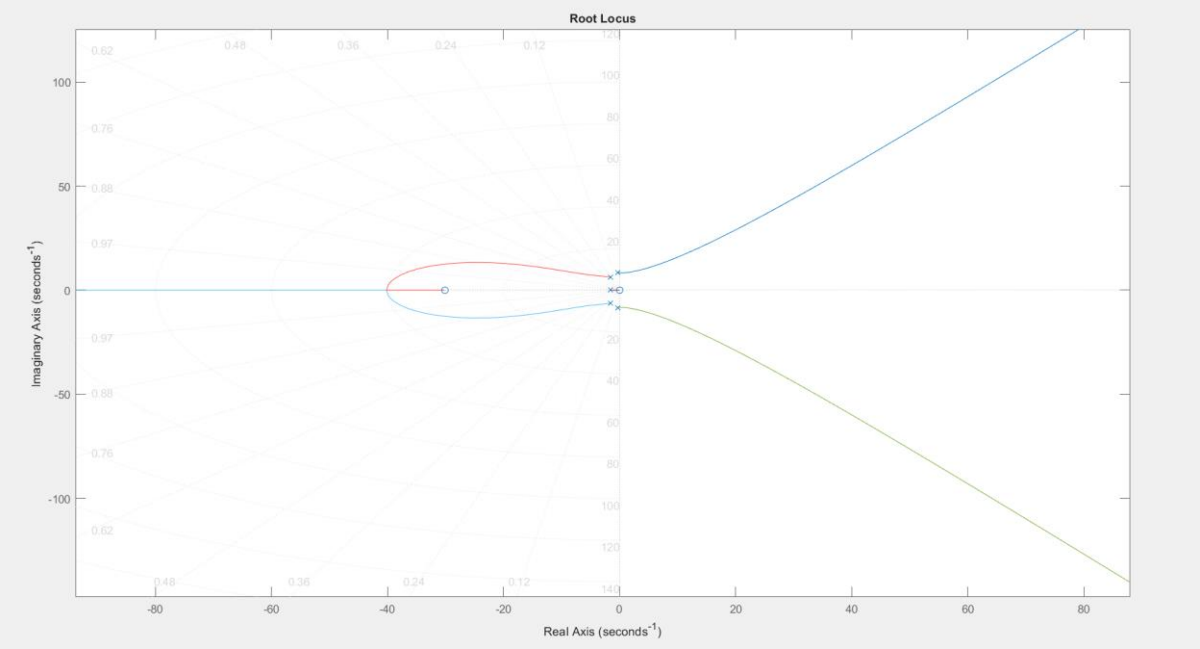


Figure 22: Root Locus for New Open Loop System

Zooming in;

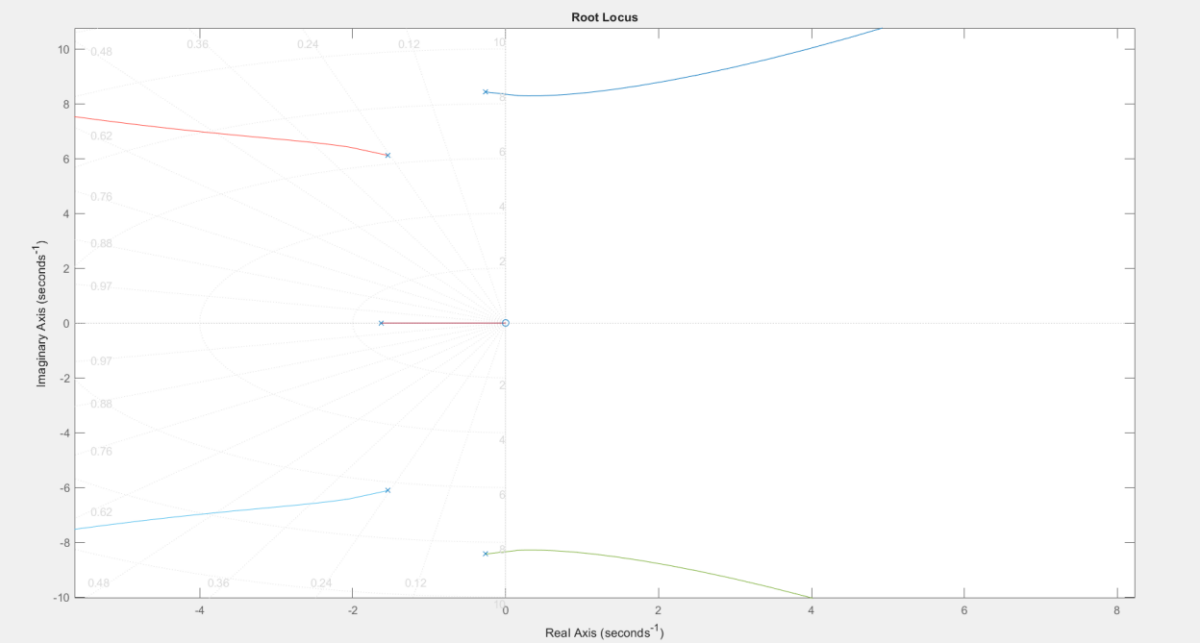


Figure 23: Zoomed in Root Locus for New Open Loop System

Run this code on terminal:
[K_selected, poles]=rlocfind(openloop_zn);

```
K_selected =

   1.4245e-04

>> poles

poles =

  -0.0045 + 8.3263i
  -0.0045 - 8.3263i
  -1.8398 + 6.3172i
  -1.8398 - 6.3172i
  -1.5313 + 0.0000i
```

Figure 24: Ku Value and Corresponding Pole Locations

$Ku = 1.4245\ x10^{-4}$ "$Ultimate\ Gain$

The ultimate gain is very small.However, it is not surprising as the poles of the plant are already very close to imaginary axis meaning that with very small gain the system will have poles on imaginary axis.

$w = 8.3263\ rad/s$ "$frequency\ of\ oscillations$"

$Pu = \frac{2\pi}{w} = 0.7546\ seconds$ "$Ultimate\ Period$"

Using the Ziegler-Nichols design table for Ku and Pu.The Dependent Ideal PID parameters are found as:

$\mathrm{Kc} = 8.3794\ x10^{-5}$ "$Proportinal\ Term$"

$\tau\mathrm{I} = 0.3773\ s$ "$Integral\ Time\ Constant$"

$\tau\mathrm{d} = 0.094325\ s$ "$Derivative\ Time\ Constant$"

2.

By looking at the transfer function of the plant it easy to see that the dominant **τ** pole pair is:

$Dominant\ pole = -0.2607 \pm 8.429i$

$\tau\mathrm{p} = \frac{1}{0.2607} = 3.8359\ seconds$ "$plant\ time\ constant$"

The delay of the system can be approximated as the sum of time constants other than the dominant pole pair.

$\mathrm{Qp} = \frac{1}{1.6263} + \frac{1}{1.5361} = 1.2659\ seconds$ "$process\ dead\ time$"

The process gain is can be found by using the estimated transfer function as follows.

$$\mathrm{Kp} = \lim_{s\to 0} Gp(s) = \frac{-190}{4596} = -0.0413 \text{ "}process\ gain\text{"}$$

Now , the desired time constant will be chosen.You can choose a more aggressive or conservative Tc apart from our choice.The moderate choice is selected for this case.

$$\tau c = Max(\tau p, 8Qp) = Max(3.8358, 10.1272\ ) = 10.1272\ seconds$$

Note that the given experimental data shows that the ball returns to its stable equlibrium point approximately 16 seconds.However with this design , it would be faster response such that 10.1272 seconds of time duration is expected to return its equlibrium point.

By using the Ideal Dependent Parameter formulation based on IMC table the parameters are found as:

$$\mathrm{Kc} = -10.056 \text{ "}Proportinal\ Term\text{"}$$

$$\tau\mathrm{I} = 4.46875\ s \text{ "}Integral\ Time\ Constant\text{"}$$

$$\tau\mathrm{d} = 0.5433\ s \text{ "}Derivative\ Time\ Constant\text{"}$$

3. The advantages of the Ziegler-Nichols Method:

- Requires no mathematical or any model of the system.
- It is an experimental method
- Easy to apply
- Effective for oscillatory systems that prone to oscillatory behaviour ,as it uses the uses the system's natural oscillation characteristics to determine tuning parameters.

- This design method represent a starting point that you can experiment with it.

The disadvantages of the Ziegler-Nichols Method:

- System is pushed at its stability limit. It may be difficult to come back stable behaviour.
- Usually it is quite undesirable for the unstable open loop plants.
- Safety issues can occur when implementing.
- It may be very time consuming if the process time constant is large.
- Decay Ratio (1/4) may be unacceptable for some applications.

21. The advantages of the FOPDT Method:

- The method is mathematically simple, making it easier to work with analytically .
- It is applicable to many world systems such as thermal, chemical,flow whose behaviour
- Captures the critical system dynamics like dominant time constant and time delay.
- It is widely used in the industry.

The disadvantages of the FOPDT Method:

- The FOPDT neglects higher order terms can lead to less accurate predictions of dynamic behaviour of the system.
- It oversimplifies the complex systems with high order dynamics or nonlinearity.
- It assumes the process is linear which is not true for many real world systems.

- It is not suitable for the oscillatory systems as the FOPDT model cannot capture oscillatory or the underdamped behaviours accurately.

22. The Ziegler Nichols Method of continuous oscillations is a more accurate PID design for this case.This is due to the fact that the system is a nonlinear high order and it exhibits oscillations. The FOPDT approximation of the plant model is not enough to catch this oscillatory dynamic behaviour as it oversimplifies the nonlinear effects and the oscillatory behaviour in the transient response. Apart from that, the estimated transfer function has poles that are very close to the imaginary axis. With a very small proportional gain, the plant can have poles on an imaginary axis. It is a known fact that for systems with poles near the imaginary axis, stability is sensitive to small changes in gain. The Ziegler-Nichols method takes this into account by tuning directly based on the system's natural frequency and damping. To conclude, Ziegler-Nichol's Method of continuous oscillations is a better choice..

Names:

Section:

Date:

5 Experimental Procedure Results

6. Root Locus:

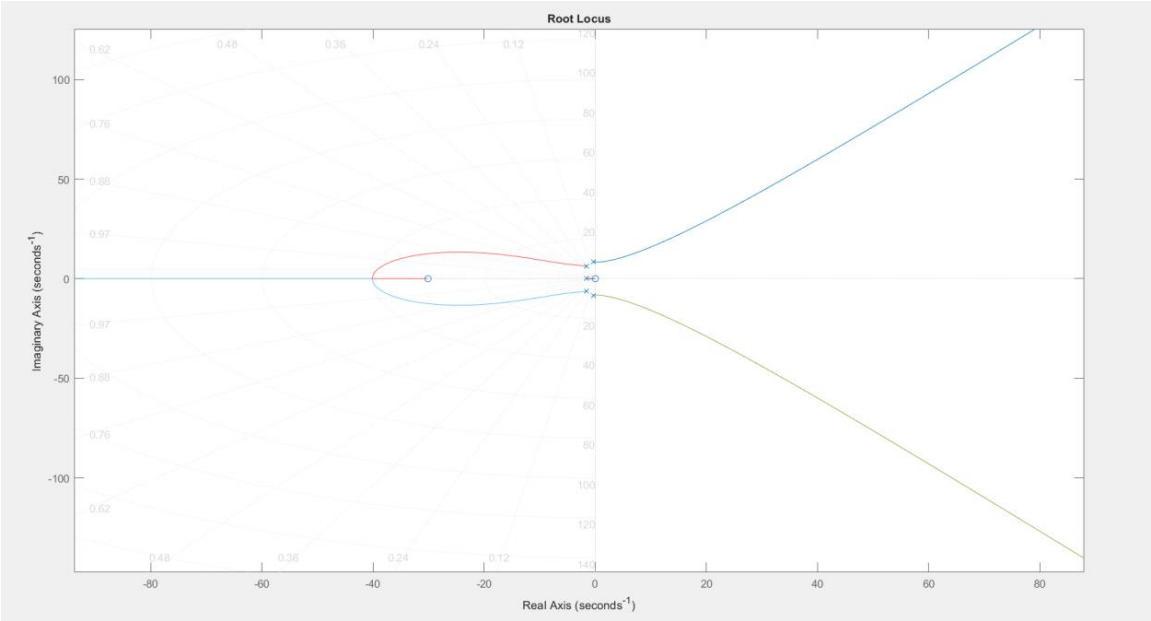


Figure 25: Root Locus for the Open Loop System

19. Table for settling time for each case**:**

Plant	10.28 seconds
P controller	To be measured (May be unstable)
PI controller	6.38 seconds
PID controller	2.97 seconds

Table 1: Settling time for Each Type of Controller

20. Optimized settling time (to be found by students) and comment:

P controller reduces the steady state error and improves the systems response in terms of being faster response.However, it increases the overshoot.

PI contoller (Effect of the Integral Action):

It completely eliminates the steady state error the type 1 system.However, it increases the settling time and overshoot such that the system becomes less stable

PID controller (Derivative action):

It decreases the overshoot and oscillations .Also, it decreases the setttling time and fastens the system. It has no effect on the steady state error.

References

[1] [Online]. Available: https://github.com/aa4cc/flying-ball-in-hoop/tree/master [Accessed: 28-Dec-2024].

[2] ETC, "MG996R Servo Motor Datasheet," *AllDatasheet*, [Online]. Available: https://www.alldatasheet.com/datasheet-pdf/download/1131873/ETC2/MG996R.html. [Accessed: 28-Dec-2024].

[3] Wikipedia, "Servomotor," *Wikipedia*, The Free Encyclopedia. [Online]. Available: https://en.wikipedia.org/wiki/Servomotor. [Accessed: 28-Dec-2024].

[4] Arduino, "Servo Library Documentation," *Arduino Documentation*, [Online]. Available: https://docs.arduino.cc/libraries/servo/. [Accessed: 28-Dec-2024].

[5] Wikipedia, "Pulse-width modulation," *Wikipedia*, The Free Encyclopedia. [Online]. Available: https://en.wikipedia.org/wiki/Pulse-width_modulation. [Accessed: 28-Dec-2024].

[6] M. Williams, "How to control servo motors with Arduino," YouTube, 15-Mar-2021. [Online]. Available: https://www.youtube.com/watch?v=GQLED3gmONg. [Accessed: 28-Dec-2024].

[7] R. Hill, "System Dynamics and Control: Module 9 - Electromechanical Systems (Actuators)," YouTube, 10-Jan-2017. [Online]. Available: https://www.youtube.com/watch?v=zxfYvGkDRTQ. [Accessed: 28-Dec-2024].

[8] Arduino, "Official Arduino Documentation," [Online]. Available: https://docs.arduino.cc/. [Accessed: 28-Dec-2024].

[9] Circuit Digest, "Serial Communication Protocols," *Circuit Digest*, [Online]. Available: https://circuitdigest.com/tutorial/serial-communication-protocols. [Accessed: 28-Dec-2024].

[10] DroneBot Workshop, "Serial Communication with Arduino - The details!," YouTube, 10-Jan-2017. [Online]. Available: https://www.youtube.com/watch?v=tpEo5AOSSkg. [Accessed: 28-Dec-2024].

[11] M. Gurtner and J. Zemánek, "Ball in double hoop: Demonstration model for numerical optimal control," *arXiv preprint arXiv:1706.07333*, 2017. [Online]. Available: https://arxiv.org/abs/1706.07333. [Accessed: Jan. 3, 2025].

[12] Wikipedia, "Lagrangian mechanics," *Wikipedia, The Free Encyclopedia*, Dec. 31, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Lagrangian_mechanics. [Accessed: Dec. 31, 2024].