Middle East Technical University

Electrical and Electronics Engineering Department

EE407 - Process Control Laboratory

Experiment 7

Ball and Hoop, PID Design

1 Objective

In this lab, you will experimentally observe and model the dynamic behaviour of a ball moving in a hoop using system identification approaches and tools. Initially, you will model the system without a controller. Later, multiple controllers will be implemented to evaluate which method is more efficient and to understand the reasons behind their performance. You will test how well analytical studies align with real-life implementations and provide insights by comparing the results. Throughout this process, you will gain a deeper understanding of the discrepancies between theoretical modelling and practical applications, refining your knowledge of system dynamics and control methods.

2 Experimental Setup

The overall experimental setup is given in Fig. 1.

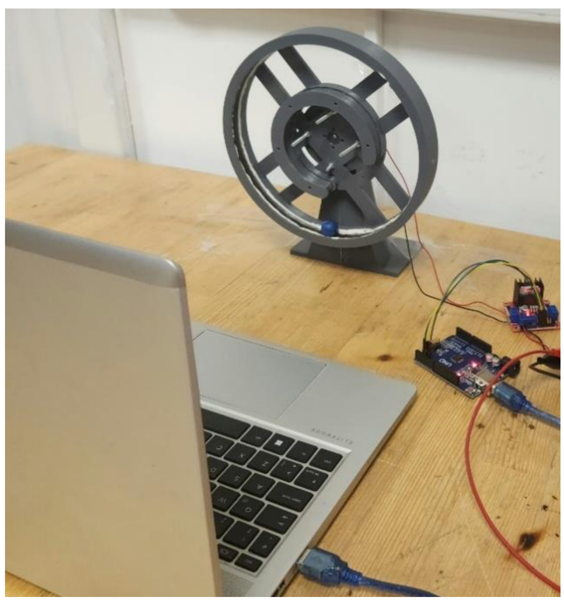


Figure 1. *Overall experimental setup*

The system consists of several interconnected components, each playing a critical role in maintaining the balance of a ball within a circular hoop. The system is shown in Fig. 2 as a block diagram.

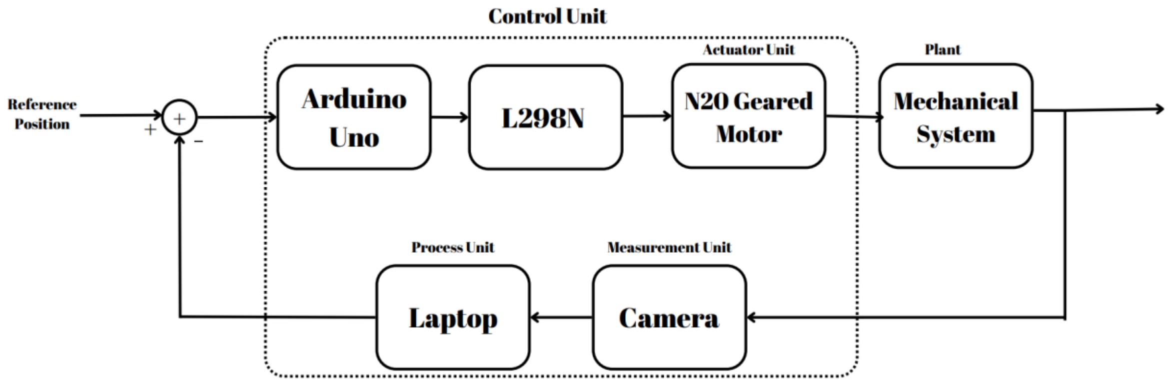


Figure 2. *Block diagram of the ball and hoop system*

Main part of measurement unit is a camera that continuously monitors the position of the ball within the hoop. The camera is used for capturing real-time visual data essential for tracking the ball's movements and detecting any deviations from the desired centre position. The laptop analyses the video feed using image processing algorithms, typically implemented through Python and OpenCV. The algorithms extract the ball's position by identifying its coordinates relative to the centre of the hoop. Any deviation from the centre is calculated as an error signal, which forms the basis for corrective action.

Once the error signal is computed, the laptop communicates this information to the control unit, consisting of an Arduino Uno microcontroller. The Arduino processes the error signal and applies a PID (Proportional-Integral-Derivative) control algorithm to determine the necessary adjustments required to minimize the error. The control signal generated by the Arduino is then transmitted to an L298N motor driver. The motor driver receives the control signals from the Arduino and converts them into the appropriate voltage and current levels needed to drive the N20 geared motor. By varying the direction and speed of the motor, the L298N driver enables precise control over the tilt of the hoop.

The N20 geared motor is the actuator unit. As the motor rotates, it tilts the hoop, influencing the movement of the ball. As the motor tilts the hoop and influences the ball's position, the camera captures the updated position of the ball, completing the feedback cycle. The wiring diagram of the system is given in Fig. 3.

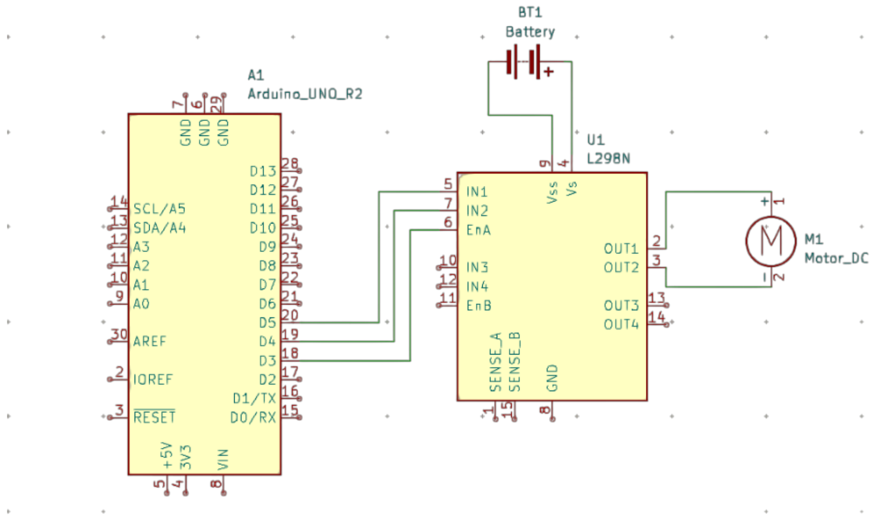


Figure 3. *Wiring diagram of the system*

3 Preliminary Work

3.1 Background Information

Originally, the ball and hoop system is used to simulate the behavior of liquids inside a moving container. This model includes a hoop with radius R and a ball with radius r and mass m as illustrated in Fig. 4. In this figure, ϑ describes the angle by which the hoop rotates, ψ represents the angular displacement of the ball, and Φ signifies the ball's angular rotation. y indicates the ball's location on the hoop's circumference, measured from a designated starting point (point A). Additionally, the ball's upward movement is considered positive along the x-axis.

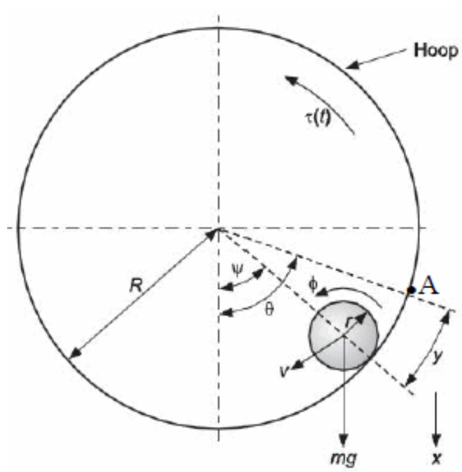


Figure 4. *Ball and hoop system illustration*

To understand how the system moves, we need equations that describe the hoop's rotation (ϑ)

and the ball's position (y) along the hoop. These two variables are used to define the system's motion as shown by (1).

$$\phi = y/r,\ v = (R-r)\psi, \quad \psi = (\theta - y/R) \tag{1}$$

Using the kinetic energies associated with the hoop's rotation, the ball's rotation, and the translational motion of the ball's center of mass, the mathematical model of the system is obtained. The movement of the system is completely defined by (2) for the hoop's rotation and (3) for the ball's position along the hoop. In these equations, I_a and I_b are the moment of inertia of the hoop and ball respectively, b_m and b_b are the rotational friction coefficient of the motor, and the friction coefficient of the ball, respectively.

$$\begin{aligned}\left[I_a + m(R-r)^2\right]\ddot{\theta} + b_m\dot{\theta} - \frac{m(R-r)^2}{R}\ddot{y} &= \dots \\ &\dots \tau(t) - mg(R-r)\sin\psi\end{aligned} \tag{2}$$

$$\begin{aligned}\left[\frac{I_b}{r^2} + \frac{m(R-r)^2}{R}\right]\ddot{y} + \frac{b_b}{r^2}\dot{y} - \frac{m(R-r)^2}{R}\ddot{\theta} &= \dots \\ &\dots mg\frac{(R-r)}{R}\sin\psi\end{aligned} \tag{3}$$

The ball and hoop system has rich and complex dynamic behaviors since it oscillates and always changes. To simplify the analysis, we can linearize equations (2) and (3) by making certain assumptions. First, we assume the slosh angle (ψ) is very small, allowing us to approximate sin(ψ) as ψ. Second, we assume the ball's rolling radius (r) is significantly smaller than the hoop's radius (R). By incorporating these simplifications, equations (2) and (3) can be rewritten in a more straightforward, linearized form as given in (4) and (5).

$$I_a\ddot{\theta} + b_m\dot{\theta} = \tau(t) \tag{4}$$

$$\left[\frac{I_b}{mr^2} + 1\right]\ddot{y} + \frac{b_b}{mr^2}\dot{y} + \frac{g}{R}y = R\left(\ddot{\theta} + \frac{g}{R}\theta\right) \tag{5}$$

Taking the Laplace Transform of (5), the transfer function of the mechanical systems is obtained as given in (6).

$$\frac{Y(s)}{\theta(s)} = R\frac{s^2 + g/R}{\left(\frac{I_b}{mr^2} + 1\right)s^2 + \frac{b_b}{mr^2}s + g/R} \tag{6}$$

3.2 Assignments

1. Find the closed loop transfer function when a PID controller is applied to the system.

$$G(s) = R \cdot \frac{s^2 + \frac{g}{R}}{\left(\frac{I_b}{mr^2} + 1\right) s^2 + \frac{b_b}{mr^2} s + \frac{g}{R}}$$

Let:

- $a_2 = \frac{I_b}{mr^2} + 1$ (coefficient of s^2),
- $a_1 = \frac{b_b}{mr^2}$ (coefficient of s),
- $a_0 = \frac{g}{R}$ (constant term).

Then:

$$G(s) = R \cdot \frac{s^2 + a_0}{a_2 s^2 + a_1 s + a_0}$$

General PID controller has the following form:

$$C(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

$$T_{\text{open}}(s) = C(s) \cdot G(s)$$

$$T_{\text{open}}(s) = \frac{K_d s^2 + K_p s + K_i}{s} \cdot R \cdot \frac{s^2 + a_0}{a_2 s^2 + a_1 s + a_0}$$

$$T_{\text{open}}(s) = \frac{R \cdot (K_d s^2 + K_p s + K_i) \cdot (s^2 + a_0)}{s \cdot (a_2 s^2 + a_1 s + a_0)}$$

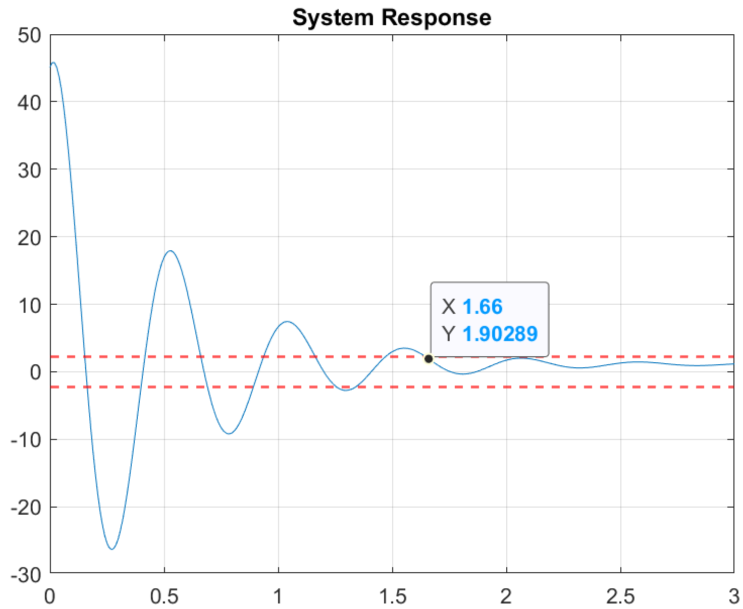
$$T_{\text{closed}}(s) = \frac{\frac{R \cdot (K_d s^2 + K_p s + K_i) \cdot (s^2 + a_0)}{s \cdot (a_2 s^2 + a_1 s + a_0)}}{1 + \frac{R \cdot (K_d s^2 + K_p s + K_i) \cdot (s^2 + a_0)}{s \cdot (a_2 s^2 + a_1 s + a_0)}}$$

$$T_{\text{closed}}(s) = \frac{R \cdot (K_d s^2 + K_p s + K_i) \cdot (s^2 + a_0)}{s \cdot (a_2 s^2 + a_1 s + a_0) + R \cdot (K_d s^2 + K_p s + K_i) \cdot (s^2 + a_0)}$$

2. The system that we are dealing with is a nonlinear system. Its natural equilibrium point is when the ball comes to the bottom of the hoop. This is a stable equilibrium since letting the ball free from any other points on the hoop will result in the object coming to this equilibrium.

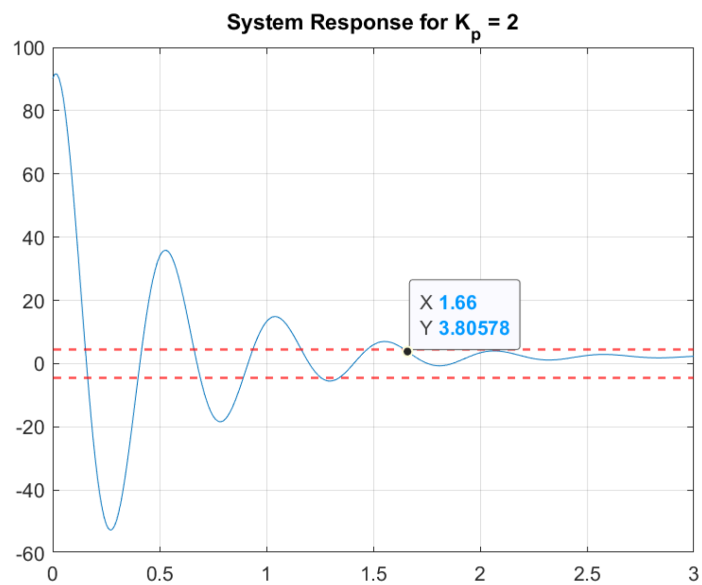
   Plot the response of the system without controller implementation and find 5% settling time. Answer the questions in preliminary work by assuming the transfer function is as given in Eq. 7.

$$f(s) = \frac{5s^2 + 30s + 19}{s^2 + 4s + 154} \tag{7}$$



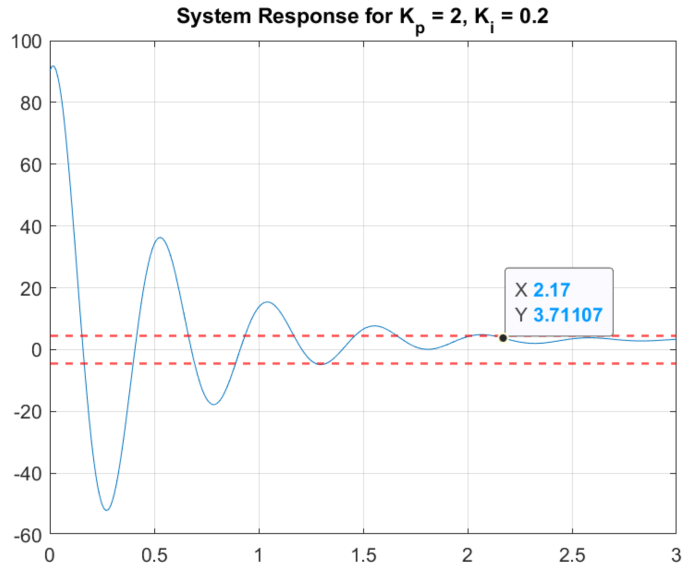
t_s = 1.66 s

3. Plot the response of the system by applying proportional controller with gain 2 and observe the settling time.



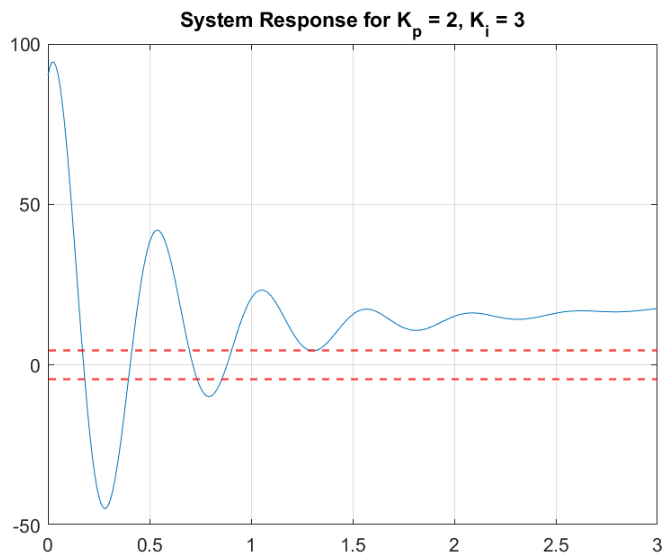
t_s = 1.66 s

4. Plot the the system's response by applying proportional controller with gain 2 and integral controller with gain 0.2. Show the 5% settling time on the plot.



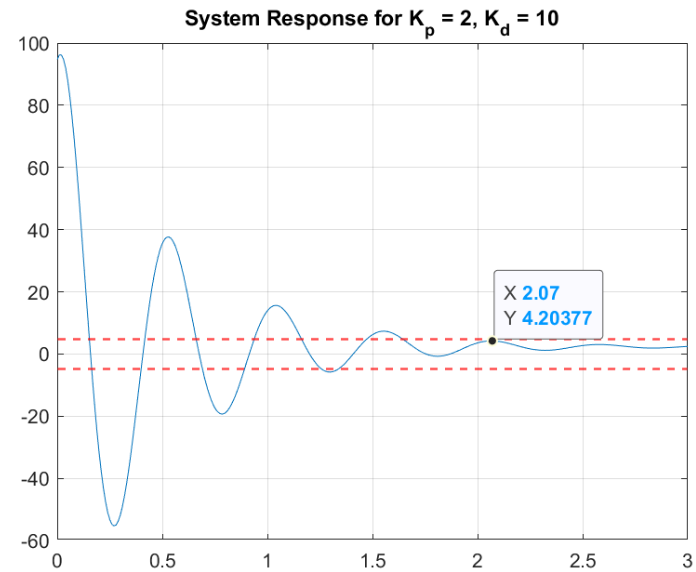
t_s = 2.17 s

5. Plot the system response by applying proportional controller with gain 2 and integral controller with gain 3. Find the settling time on the plot.



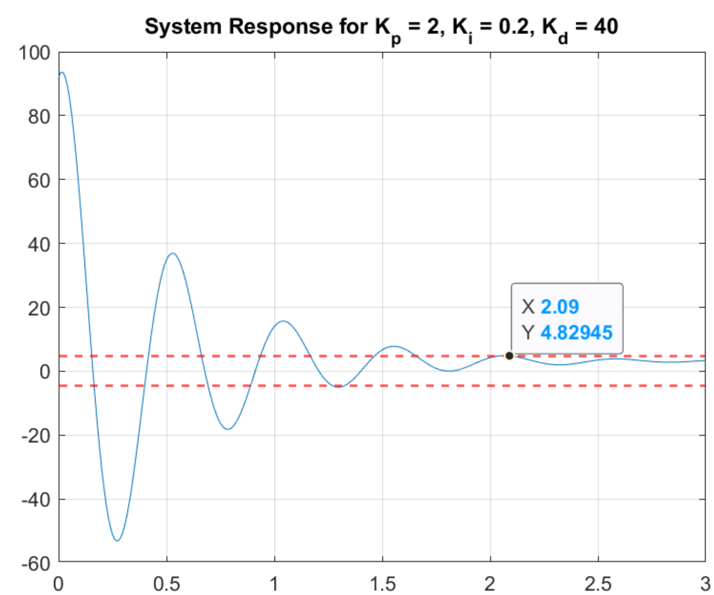
The system does not reach 5% settling band since the oscillations of the system increases due to high K_i value.

6. Plot the response of the system by applying proportional controller with gain 2 and differential controller with gain 10. Show the 5% settling time on the graph.



t_s = 2.07 s

7. Plot the response of the system by applying PID controller with K_p=2, K_i=0.2 and K_d =40. Show the 5% settling time on the graph.



t_s = 2.09 s

8. Comment on the results. Which controller can be ideal to control the ball and hoop system?

A proportional controller generates an output that is directly proportional to the current error. This term helps to reduce the error by applying a correction that is proportional to the magnitude of the error; however, higher proportional gain can cause overshooting and instability. An integral controller addresses accumulated past errors by summing the error over time. On the other hand, too much integral action can lead to an excessive correction, resulting in overshoot or oscillations. A derivative term anticipates future errors by considering the rate of change of the error. It dampens the system's response by applying corrective action based on how fast the error is changing.

We want the system to reach steady state as fast as possible with minimum steady state error. Therefore, it is expected that PID controller gives the efficient solution for this experiment.

4 Experimental Work

1. Open the exp.ino file. Inside the file, a PID controller is implemented. Set proportional gain, Kp, to 2. Integral and derivative gains, K_i and K_d, should be set to 0. Verify the file with the button on the top left. Then upload it to the Arduino by using the button next to it. Keep the DC Power Supply off for now.
2. Run the Python file ball.py and start recording the movement. The setup should be well illuminated, and the ball color should be tracked with the camera. The script prints the Ballpos_x value which should be 0 at the equilibrium point in the correct camera position, if not adjust so. Power up the motor controller by turning on the DC Power Supply. When stopped, script will output a .mat file in order to process in the MATLAB in the later steps.
• The system has a dead-band around the equilibrium point. You might consider the reasoning behind and the use of it.
3. Import the data.mat file into the MATLAB and parse it so that only the response is captured. Calculate the sampling time using your time data in MATLAB.
4. Run “System Identification Tool” from the Apps tab of the MATLAB. Click “Import data” and select “Time Domain Resp.”. Select import variables. Starting time should be set to 0 and enter the sampling time you have calculated in the previous step.
5. Select “Estimate” then select “Transfer Function Models..”. Enter the number of poles and zeros and set “Initial Conditions” to “Estimate”. Click “Estimate”. Check the performance of the estimation and note the transfer function. Calculate the settling time in the %5 band using your response figure. Put your figures and findings in the report.
6. Repeat steps 1-4 with Kp=2, Ki=0.2 while Kd is set to 0. Don’t forget to power on after adjusting the equilibrium point. Save the figures and transfer function. Comment on the findings and PI response of the system.
7. Repeat the PI response for Kp = 2 and Ki=3. Comment on the oscillatory response. Mention the reasoning behind the oscillations.
8. Design a PD controller by repeating the steps with Kp=2 and Kd =10, while keeping the Ki at zero. Record the findings and comment on the PD response of the system.
9. For PID response set Kp=2, Ki =0.2 and Kd =40 and repeat the procedure. Comment on the PID response.
10. Which is the best system to stabilize the ball? Finalize your comments.

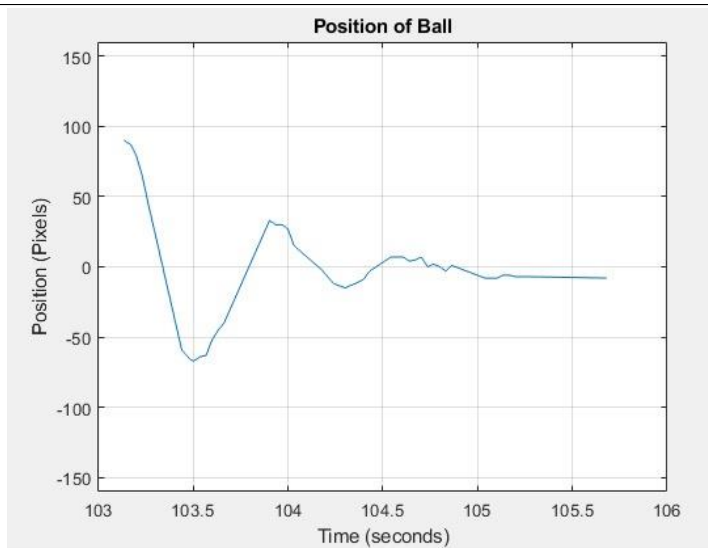
Names: Bora Özkan 2516748
Elifnur Balaban 2515591
Emirhan Yolcu 2517332

Section:

Date: 02.01.2025

6 Experimental Procedure Results

1. P response of the system (*pixels vs time)*



2. Settling time, t_s

$$t_s(\%) = 104.82 - 103.13 = 1.69\,s$$

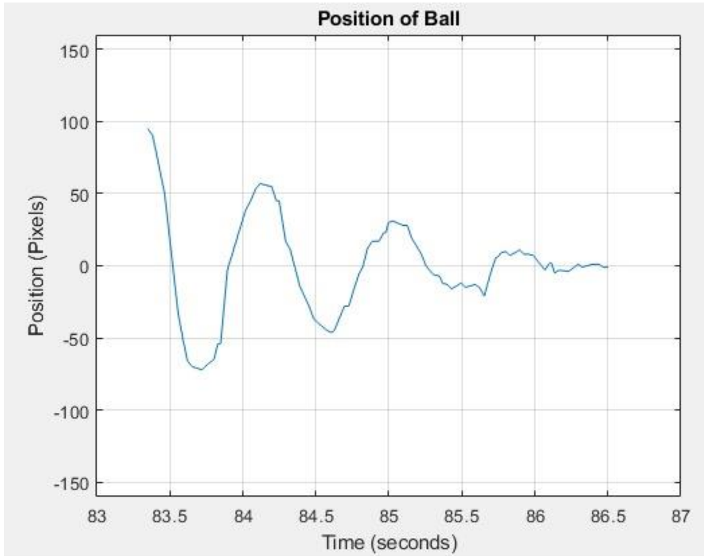
3. Transfer function of the P-Response, T(s)

$$T(s) = \frac{20.93s^2 + 50.51s - 2.536}{s^3 + 8.029s^2 + 76.78s + 223.6}$$

4. Comments on the P-Response

With P-Controller implementation, we have observed a faster response to the disturbance. The settling time has decreased compared to the plant response. Therefore P-Controller would be a better choice in terms of response time. Also, normally, P-Control causes a steady state error. This error is eliminated by applying a dead-band. Therefore ball will always stay within ±15 pixel band which is accepted as our equilibrium band by design.

5. PI response of the system (*pixels vs time)*



6. Settling time, t_s

$$t_s = 85.89 - 83.34 = 2.55\ s$$

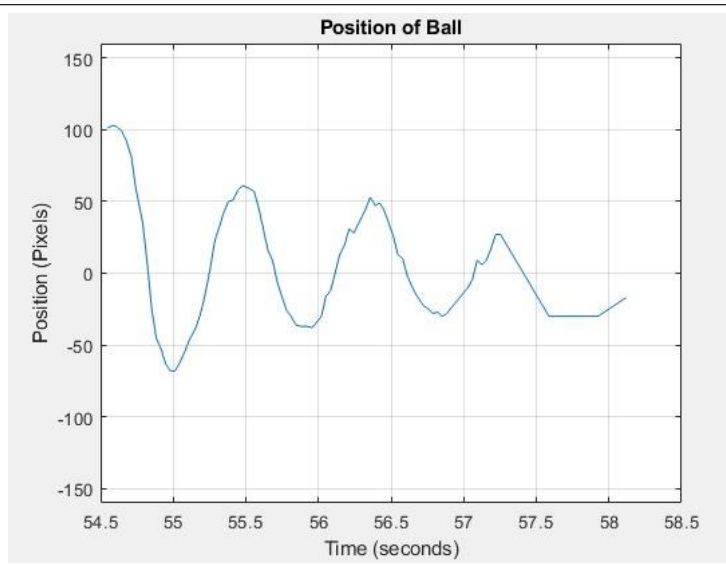
7. Transfer function of the PI-Response, T(s)

$$T(s) = \frac{1.136s^3 - 62.54s^2 + 199.7s - 48.75}{s^3 + 48.25s^2 + 149.9s + 2702}$$

8. Comments on the PI-Response

With I term included, we observe some oscillations. This is a result of the overshoot behavior mostly caused by the I term. The I increases our response speed but also introduces some unstability. I-Control is mostly used to eliminate steady state error. Since this system always converges to equilibrium, it is not needed. Hence, the PI-Controller is not a good option.

9. PI response of the system with over-oscillatory behavior (*pixels vs time)*



10. Settling time, t_s

It did not reach settling (5%) band during the measurement because of the controller's oscillatory behavior.

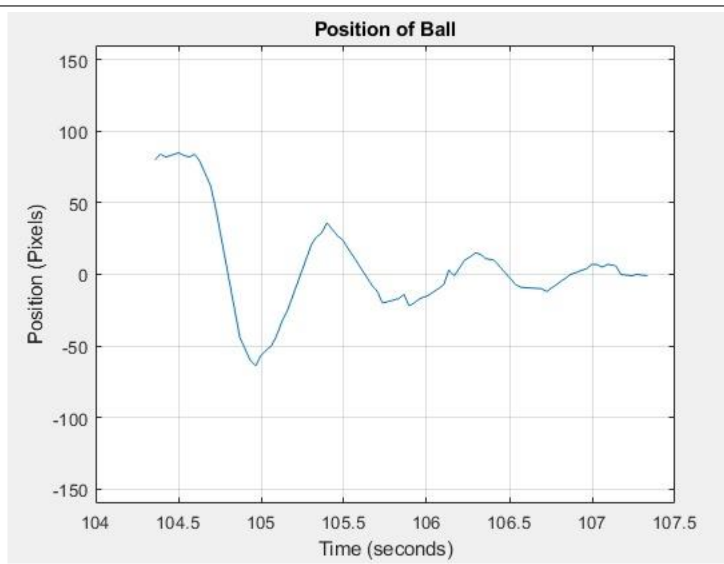
11. Transfer function of the new PI-Response, T(s)

$$T(s) = \frac{406.6s^2 + 1581s + 601.5}{s^4 + 6.358s^3 + 391s^2 + 481.4s + 1.08\times10^4}$$

12. Comments on the oscillatory PI-Response

Increasing I-term causes increasing oscillations. As I-term increases, it will result in a behavior similar to the undamped response. Further increase may push the system to the unstable region.

13. PD response of the system (*pixels vs time)*



14. Settling time, t_s

$$t_s = 107.05 - 104.5 = 2.55\,s$$

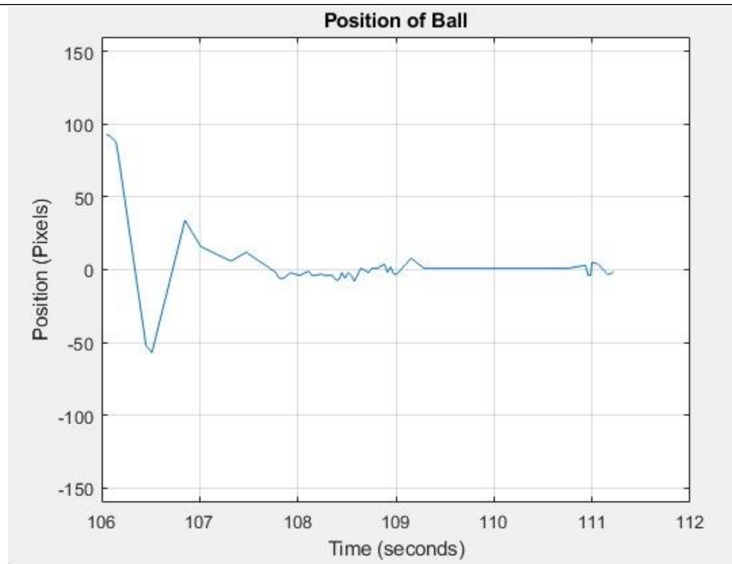
15. Transfer function of the PD-Response, T(s)

$$T(s) = \frac{238.1 s^2 + 215.8 s - 352.9}{s^3 + 273.3 s^2 + 9275 s + 1.553 \times 10^4}$$

16. Comments on the PD-Response

We observe that D-Control suppresses the controller response. Therefore, the settling time is increased.
In theory, D term should allow expanding the use of other terms.

17. PID response of the system (*pixels vs time)*



18. Settling time, t_s

$$t_s = 107.8 - 106.15 = 1.65\,s$$

19. Transfer function of the new PID-Response, T(s)

$$T(s) = \frac{142.8s^3 + 535.8s^2 + 1.365\times10^4 s - 115.3}{s^4 + 12.61s^3 + 355.6s^2 + 2424s + 1.837\times10^4}$$

20. Comments on the oscillatory PID-Response

In this control method, we achieved the equilibrium point with a quick settling time. The use of D-Control allowed us to employ integral control. With that, the settling time has decreased drastically. We have a settling time similar to P-Control but with less oscillations.

21. Final comments:

In this experiment, we tried to achieve the fastest settling. Since there is a dead-band applied and the plant naturally converges to equilibrium, the steady state error was not our concern. We observed effects of different control terms and designed a PID-Controller that results in an optimized response time.

Appendices

- aa4cc. (2017, June 30). Ball and Hoop - motivation for numerical optimal control [Video]. YouTube. https://www.youtube.com/watch?v=484GN4KBQnc
- Aa4cc. (n.d.). GitHub - aa4cc/flying-ball-in-hoop: Codes, models, sketches and everything other associated with the demonstration model "Flying ball in a hoop" developed at CTU in Prague. GitHub. https://github.com/aa4cc/flying-ball-in-hoop
- Aa4cc. (n.d.-a). Flying ball in hoop. https://aa4cc.github.io/flying-ball-in-hoop/
- Calvo, I., Zulueta, E., Oterino, F., & López-Guede, J.M. (2009). A Remote Laboratory for a Basic Course on Control Engineering. *Int. J. Online Eng., 5*, 8-13.
- Gurtner, M., & Zemánek, J. (2017, June 22). Ball in double hoop: demonstration model for numerical optimal control. arXiv.org. https://arxiv.org/abs/1706.07333
- Kanthalakshmi, s & V.MANIKANDAN,. (2010). Genetic Algorithm Based Self Tuning Regulator. International Journal of Engineering Science and Technology. 2.
- Virtual and remote experimentation with the Ball and Hoop system. (2009, September 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5347108