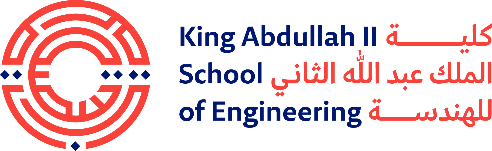
Princess Sumaya University for Technology

King Abdullah II Faculty of Engineering

Electrical Engineering Department



|  |
| --- |
| **Embedded Systems Lab 22448**  **Final Project Report**  **Model-Based PID Ball Balancing System** |

|  |  |  |  |
| --- | --- | --- | --- |
| *Authors:* |  |  | *Supervisor:* |
| Laith AlKhdour  Omar Abujumaa  Tarek Salameh | 20190130  20190561  20190217 | Computer Engineering  NIS Engineering  Computer Engineering | Dr. B. Sababha  Eng. R. Al-Harasis |

*June 03, 2023*

***Abstract***

*The Proportional-Integral-Derivative (PID) controller is being developed as part of the Model-Based PID Ball Balancing System project in order to create a reliable control system that can keep a ball balanced on a surface. The system uses servo motors as actuators to change the position of the ball by bending the surface horizontally and ultrasonic sensors to measure how far the ball is from the desired position. To enhance the performance of the system, the values of the PID controller parameters KP, KI, and KD are manually adjusted using a potentiometer. The results of numerous experiments and investigations revealed that the values for KP, KI, and KD that produced the best performance were 1.99, 0.287, and 0.0446 respectively.*

**Table of Contents**

[**Introduction** 2](#_Toc136799299)

[Objectives 2](#_Toc136799300)

[Theory 2](#_Toc136799301)

[Equations Guidelines 2](#_Toc136799302)

[**Procedure and Methods** 4](#_Toc136799303)

[**Results and Discussions** 7](#_Toc136799304)

[**Conclusions** 7](#_Toc136799305)

[**References** 8](#_Toc136799306)

**Introduction**

The Model-Based PID Ball Balancing System is an innovative project with the aim of developing an adequate control system that is capable of maintaining a ball balanced on a surface by employing the Proportional-Integral-Derivative (PID) control algorithm.

In this project, we'll examine the key concepts behind PID controllers and place them into play with a ball-balancing system. To improve the performance of the system, the values of the PID controller parameters KP, KI, and KD are manually modified using a potentiometer.The control system will constantly bend the surface using a servo motor based on the position of the ball moving above that surface and mitigate any errors that affects its balance through the use of inputs from an ultrasonic.

In order evaluate the performance of the ball balancing system in real-world situations, we will also build a physical prototype of the system. We aim to improve the controller's accuracy and ability to sustain precise balance even in disturbances through iterative evaluation and experimentation. The Model-Based PID Ball Balancing System's successful implementation will provide evidence of the PID control algorithm's effectiveness in a real-world application and offer significant insights into the design and optimization of control systems for dynamic and unstable systems.

Objectives

* Balancing a ball on a surface using the PID controller
* The ability to tune the values of kP, kI, and kD to match the desired performance of the controller
* Deploying a PID controller into real-life projects.

Theory

Proportional-Integral-Derivative controller, or PID controllers, is a ubiquitous type of feedback control mechanism in engineering and automation systems that is intended to periodically monitor and alter a system's output with the objective to accomplish and maintain a desired setpoint or target value. Three control actions—proportional (P), integral (I), and derivative (D)—are combined by the PID controller. Based on the difference between the the target point and the system's determined output, every part makes a contribution to the total control signal [1].

Equations Guidelines

In a PID controller, there are 4 basic terms that needs to be addressed:

1. Proportional (P)

The P-Gain is multiplied by the error to determine the proportional. The proportional's objective is to have a significant immediate effect on the output and bring the process value relatively close to the set point. The proportional value's impact on the output decreases as the error grows smaller.

In this project, KP is responsible for contributing to how fast the ball reaches the desired position.

This is how proportional calculations looks like:

P = Proportional

kP = Proportional Gain

SP = Set point

PV = Process Value

Err = Error

Err = SP – PV

P = kP x Err

1. Integral (I)

The I-Gain is multiplied by the error to calculate the integral, which is then multiplied by the controller's cycle time (how frequently the controller performs the PID calculation) and continuously accumulated as the "total integral".

More specifically, the new calculated integral value is added to the total integral value each time the controller performs the PID calculation. The integral typically has less of an immediate impact on the output than the proportional, but because it continuously builds up over time, the longer it takes for the process value to get to the set point, the more of an impact it will have.

In this project, KI controls the steady state error i.e. the difference between the actual value and desired value.

The Integral math:

I = Integral

kI = Integral Gain

dt = cycle time of the controller

It = Integral Total

I = kI x Err x dt

It = It + I

1. Derivative (D)

The derivative is obtained by dividing the D-Gain by the process value's ramp rate. With the aim to hopefully avoid the controller from overshooting the set point if the ramp rate is too fast, the derivative's goal is to "predict" where the process value is going and bias the output in the opposite direction of the proportional and integral.

In a slightly more straightforward explanation, the derivative will limit the output if the process value is approaching the set point too quickly in order to stop it from overshooting the target value.

In this project, KD is adjusted to deal with any overshoot (when the desired value is higher than the actual value) or undershoot (when the desired value is lower than the actual value).

The Derivative Math:

D = Derivative

kD = Derivative Gain

dt = cycle time of the controller

pErr = Previous Error

D = kD x (Err – pErr) / dt

Note that the above equations have been derived from [1]. Figure 1 shows a controller that uses PID algorithm.

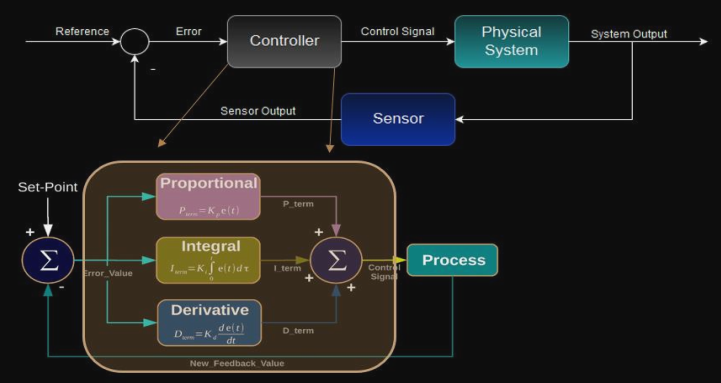


Figure PID controller

**Procedure and Methods**

This project used different hardware components and different creative ideas have been applied to come up with an innovative hardware design that matches the required physical structure. An ultrasonic sensor is used to measure the distance of the ball from the edge, and the servo motor bends the surface the ball is moving on. Three potentiometers are used for adjusting the values of P, I, and D manually to match the required functionality of the controller. Figure 2 shows how the software code is designed. MATLAB Simulink is used to build the software part of this project. Figure 3 shows the serial connection with the PID controller.

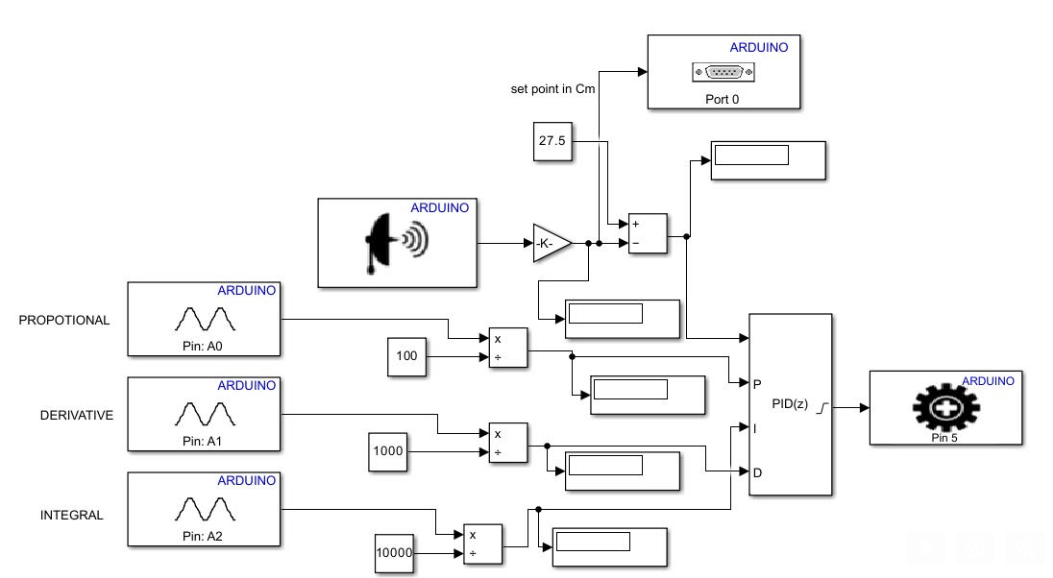


Figure MATLAB simulink file containing the PID controller

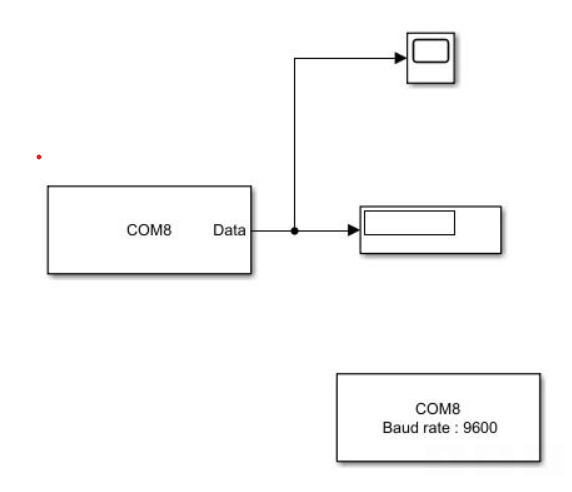


Figure Serial communication

An Arduino Uno-based circuit is built to control the physical structure of the ball-balancing system. Figure 4 shows the circuit that was built.

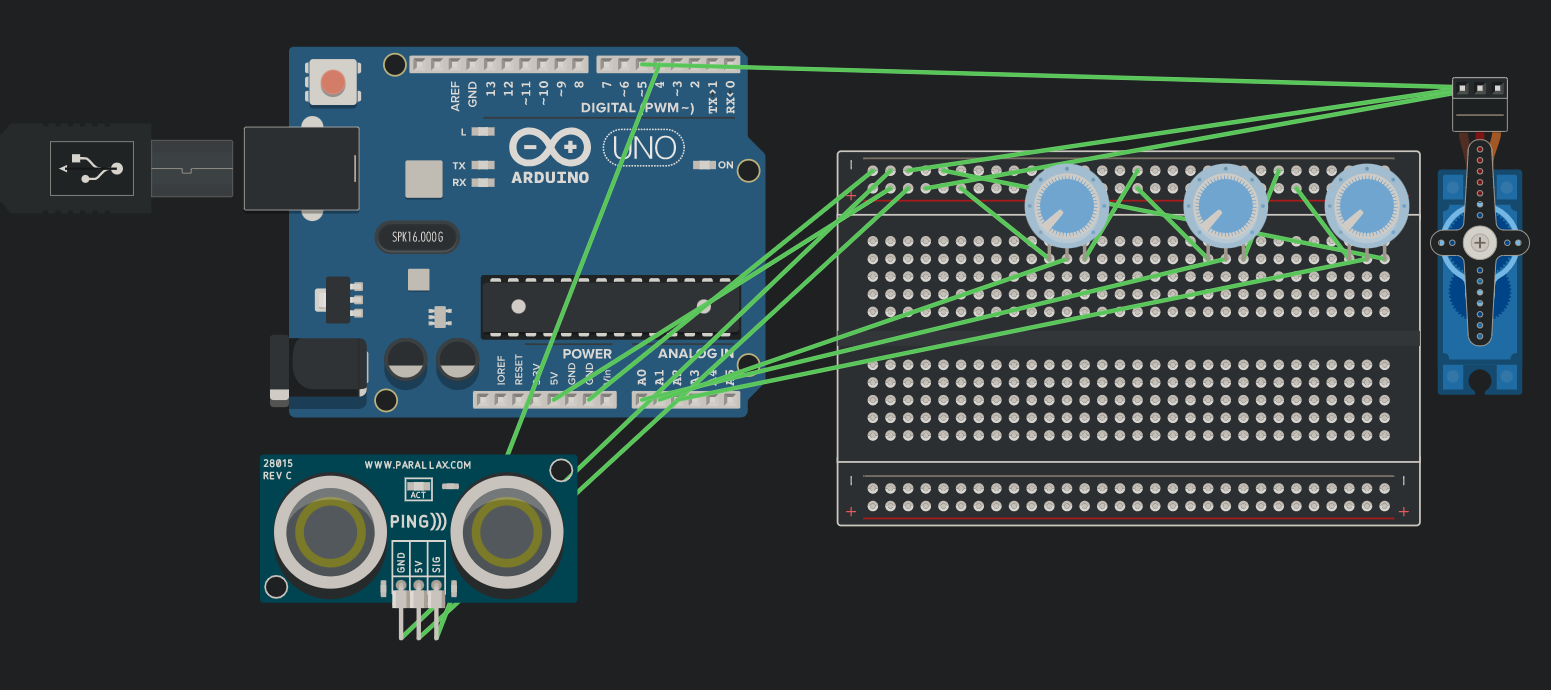


Figure Circuit controlling the system

To test the efficiency of the controller, a physical prototype was built. The prototype mimics the desired ball-balancing. Figure 5 shows the prototype built.

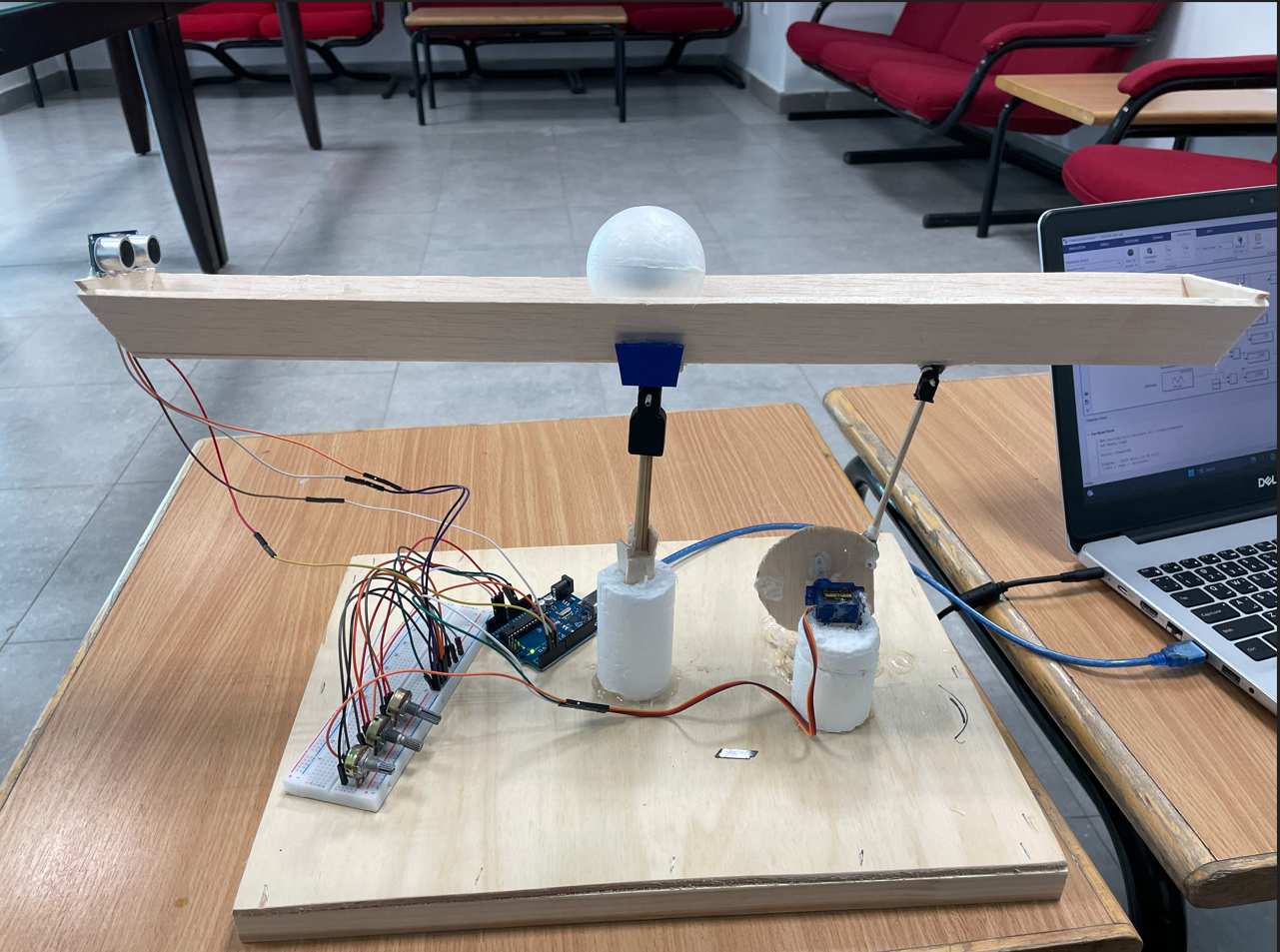


Figure Prototype

**Results and Discussions**

After several trials to tune the values for KP, KI, and KP, the controller kept the ball at the center with the values of 1.99, 0.287, and 0.0446 respectively. However, different problems occurred when the controller kept bending the surface to balance the ball. Due to the light-weight of the wood used, the rapid movement of the servo motor to the left and right lead to the ball falling out of track. Moreover, this type of movements bends the surface the ball is moving on with a particular angle that would affect the ultrasonic’s readings, making the servo motor behaves undesirably. Figure 6 shows how the distance of the ball kept changing as the servo motor kept rotating the surface.

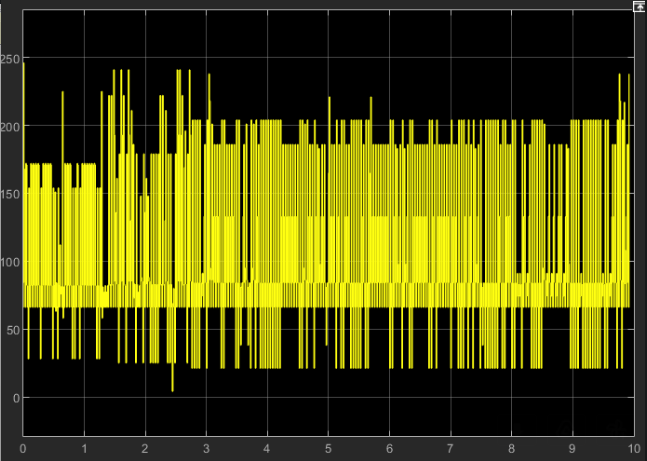


Figure The serial receiver output

**Conclusions**

As part of the Model-Based PID Ball Balancing System project, the Proportional-Integral-Derivative (PID) controller is being developed in order to develop a reliable control system that can keep a ball in balance on a surface. The device uses ultrasonic sensors to measure how far the ball is from the target position and servo motors to bend the surface horizontally in order to adjust the ball's position. By manually adjusting the values of the PID controller parameters KP, KI, and KD using a potentiometer, the system's performance can be further improved. Numerous tests and investigations brought in results showing that the optimal KP, KI, and KD values were 1.99, 0.287, and 0.0446, respectively.

**References**

|  |  |
| --- | --- |
| [1] | "PID Explained," PID Controller Explained, [Online]. Available: https://pidexplained.com/pid-controller-explained/. |