**Abstract:**

A “Ball Balancing Platform” is a dynamic control system that is designed to stabilize a ball by adjusting the orientation of the platform. A 4-wire resistive touchpad is used as the platform to obtain position feedback. The platform is mounted on a 3RRS (Revolute-Revolute-Spherical) parallel manipulator to control the orientation of the platform. A PID (Proportional-Integral-Derivative) controller is used to maintain the ball at a target position by constantly adjusting the platform’s angle based on the displacement. The main objective of the thesis is to study control strategies, electronics, inverse kinematics, sensor integration and system modelling.

**Streszczenie:**

“Platforma do balansowania kulką” to dynamiczny system sterowania, który ma za zadanie stabilizować kulkę poprzez dostosowywanie orientacji platformy. Do uzyskania informacji o pozycji wykorzystuje się 4-przewodową rezystancyjną nakładkę dotykową. Platforma jest zamontowana na równoległym manipulatorze o 3 stopniach swobody (3 DOF), który kontroluje jej orientację. Regulator PID (proporcjonalno-całkująco-różniczkujący) jest używany do utrzymania kulki w wybranej pozycji, stale dostosowując kąt platformy na podstawie przemieszczenia kulki. Głównym celem pracy jest zbadanie strategii sterowania, elektroniki, kinematyki odwrotnej, integracji czujników oraz modelowania systemu.

Table of Contents

[1 Introduction 3](#_Toc186299961)

[2 Objectives, Assumptions and Requirements 4](#_Toc186299962)

[2.1 Objectives 4](#_Toc186299963)

[2.2 Performance Metrics 4](#_Toc186299964)

[2.3 Assumptions 4](#_Toc186299965)

[2.4 Design Constraints 5](#_Toc186299966)

[2.5 Requirements 5](#_Toc186299967)

[3 Mechanical Configuration 6](#_Toc186299968)

[4 Electrical Configuration 7](#_Toc186299969)

[4.1 Arduino UNO 7](#_Toc186299970)

[4.2 TMC2209 V2.0 8](#_Toc186299971)

[4.3 CNC Shield 9](#_Toc186299972)

[4.4 Stepper Motor 10](#_Toc186299973)

[4.5 Resistive touchpad 11](#_Toc186299974)

[5 System Modelling 12](#_Toc186299975)

[5.1 PID Controller 12](#_Toc186299976)

[5.2 Inverse Kinematics of the system 14](#_Toc186299977)

[5.3 Equations of Motion for the ball 17](#_Toc186299978)

[6 Constructing the Ball Balancer 19](#_Toc186299979)

[7 References 20](#_Toc186299980)

s

# Introduction

“Ball balancing platform” is a unique engineering problem that requires combining mechanical design, electronics, microcontroller programming and advanced control algorithms. The aim of the project is to design and build a platform that can balance a spherical metal ball at a pre-defined point by tilting the platform to counteract external forces.

The interdisciplinary nature of this project inspired me to pursue it. The integration of mechanics, electronics, and control theory makes it an ideal dissertation topic for Mechatronics. Traditionally a stereo camera is used to capture images and use image processing techniques to obtain position feedback. However, in this project a resistive touchpad is used to obtain position feedback. This is more versatile and accurate than the former method.

The practical applications of such a control algorithm extends to various fields, including but not limited to Robotics, Aerospace and Industrial Automation.

A blue and yellow robotic device

Description automatically generated

Figure 1.1 CAD Model of a ball balancing platform

In addition to building the physical prototype a MATLAB Simulink model will be used to simulate the same. This can be helpful in ensuring the validity of the inverse kinematics model and PID controller.

# Objectives, Assumptions and Requirements

This chapter outlines the objectives of the project in a systematic order of implementation. There are some research-oriented objectives as well for which the results cannot be anticipated and hence are not considered for measuring performance metric. Requirements for both hardware and software development are listed as well.

## Objectives

The objectives of the project are as follows:

1. **Construction:** Fabricate and assemble the platform successfully.

2. **Balancing:** Achieve precise balancing of a metallic ball on the platform with PID control.

3. **Simulation:** Develop a Simulink model to simulate the system.

4. **Drawing Curves:** Implement control to enable the ball to draw predefined figures on the platform.

5. **Research**: Conduct research of other control strategies that can be employed.

## Performance Metrics

The outcome of the work and the system’s success will be measured by the following metrics:

1. **Stability**: The time taken to stabilize the ball after an external disturbance.
2. **Precision**: The accuracy of the ball's position control within a predefined error range.
3. **Speed**: The system's response time to input changes or disturbances.
4. **Drawing Accuracy**: The deviation between the predefined figure and the actual path traced by the ball.

## Assumptions

This section outlines the assumptions made in work. As the Ball Balancing Platform is a demonstration built in a lab environment, the required assumptions are not very rigid. The following assumptions were made:

1. The platform’s operation is intended to be demonstrated in a lab environment.
2. A PID controller will be used primarily. Similar control strategies may be explored.
3. No Friction is considered.
4. The ball that is to be balanced is perfectly spherical and has a smooth surface.
5. The ball does not slide on the platform or moves upwards.
6. The ball in static conditions is unstable even without external disturbances.
7. Vibrations and jerky motion are not considered for simulation.
8. The speed of actuation (the angular velocity of motors) is not taken into consideration for simulation.

## Design Constraints

### Budget Constraints: As the work is self-financed, it operates within a limited budget, prioritizing cost-effective components.

### Size Limitations: The platform dimensions are restricted to ensure it is portable and can fit within the lab workspace.

### Material Restrictions: 3D-printed parts are used, limiting the material options to those compatible with the available 3D printer. PETG filament is used.

### Load Capacity: The system is designed for balancing a single steel ball, restricting its use for larger or multiple objects.

## Requirements

### Electronics

1. [Maker Uno Microcontroller](https://www.amazon.com/PJRC-Cortex-M7-Processor-iMXRT1062-Without/dp/B088JY7P2H/ref=sr_1_4?crid=36CI81XOWM0Z0&keywords=teensy+4.1&qid=1677474562&s=electronics&sprefix=teensy+4.1%2Celectronics%2C139&sr=1-4)
2. [Nema 17 59 Ncm Stepper Motors (Bipolar)](https://www.amazon.com/dp/B00PNEQKC0?psc=1&ref=ppx_yo2ov_dt_b_product_details)
3. [TMC2209 V2.0 Stepper Motor Drivers](https://www.amazon.com/dp/B082LSQWZF?psc=1&ref=ppx_yo2ov_dt_b_product_details)
4. [CNC](https://www.amazon.com/gp/product/B08BWPGSSC/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1) shield
5. [30V Bench Power Supply](https://www.amazon.com/gp/product/B082FV1PGP/ref=ppx_yo_dt_b_search_asin_title?ie=UTF8&psc=1) (or any 24V power supply)
6. [5V Regulator](https://www.amazon.com/Weewooday-Regulator-Voltage-Converter-Transformer/dp/B08JZ5FVLC/ref=sr_1_2_sspa?crid=28M4EAWMIWBMT&keywords=5v+regulator&qid=1684300783&sprefix=5v+regulator%2Caps%2C198&sr=8-2-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUExSThTTFJLVFozQ0hKJmVuY3J5cHRlZElkPUExMDMyNTI2WEZWTUZFWkhZQVNNJmVuY3J5cHRlZEFkSWQ9QTAxNTE0MDUyUVZGRzBRTkxZMUpBJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==)

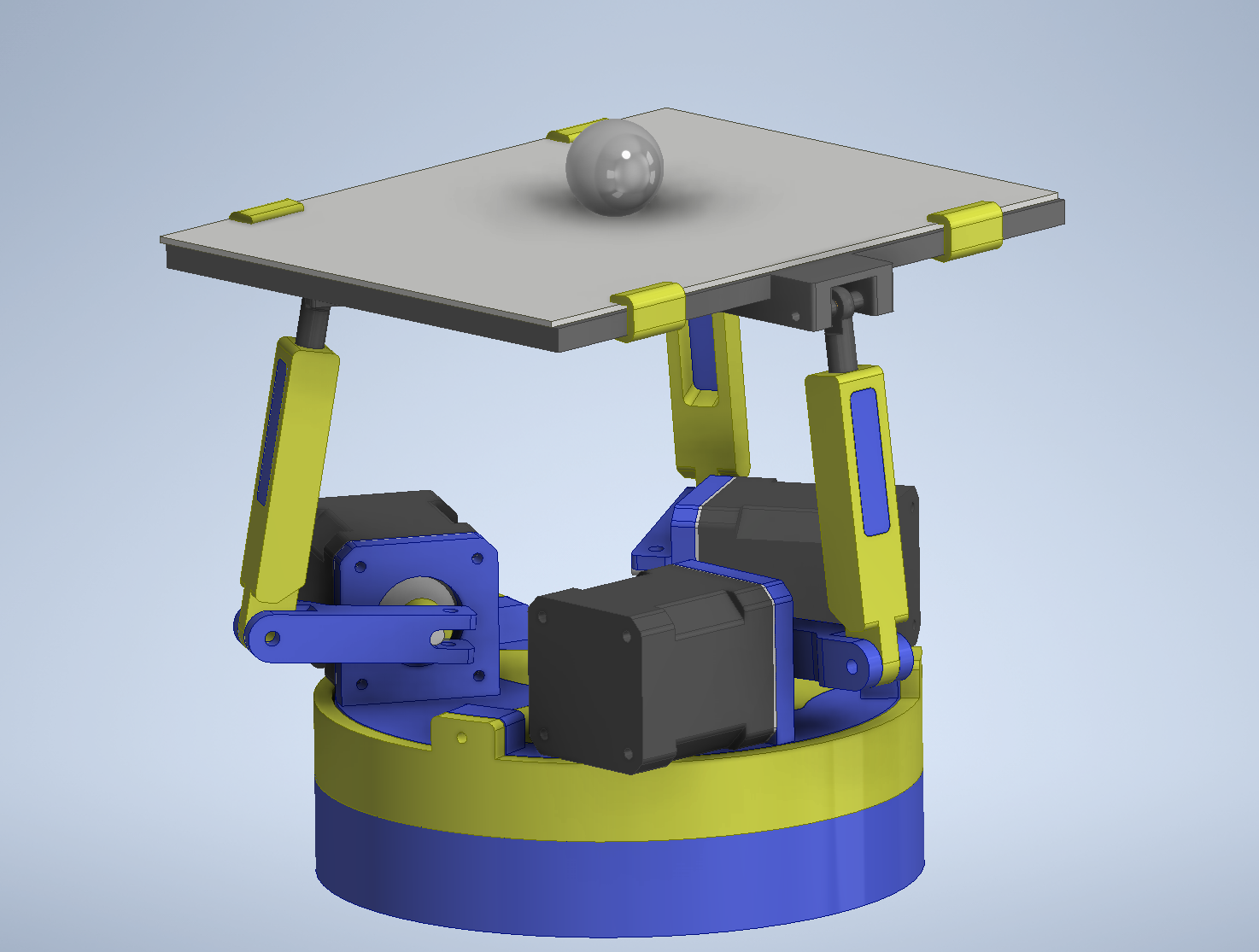
### General Parts

1. [8.4" 4 Wire Resistive Touch Panel](https://www.amazon.com/dp/B07TZGVY8K?psc=1&ref=ppx_yo2ov_dt_b_product_details)
2. [1" Steel Bearing Ball](https://www.amazon.com/dp/B07CHZ94W9?psc=1&ref=ppx_yo2ov_dt_b_product_details)
3. [22mm long m3 tie rod](https://www.amazon.com/dp/B09JLKLK73?psc=1&ref=ppx_yo2ov_dt_b_product_details)
4. [M3 x 6mm threaded inserts](https://www.amazon.com/dp/B07LBQRYR3?psc=1&ref=ppx_yo2ov_dt_b_product_details)
5. [M3 x 5mm Standoffs](https://www.amazon.com/dp/B07M7D9PRM?psc=1&ref=ppx_yo2ov_dt_b_product_details)
6. [M3 x 5mm Screws](https://www.amazon.com/Dahszhi-Phillips-Machine-Metric-Thread/dp/B099ZK3NK9/ref=sr_1_5?crid=18J5PGUV1ZTGU&keywords=m3%2Bx%2B5mm%2Bscrews&qid=1681514624&sprefix=m3%2Bx%2B5mm%2Bscrews%2Caps%2C110&sr=8-5&th=1)
7. [M3 x 8mm Screws](https://www.amazon.com/Socket-Screws-Metric-Stainless-Machine/dp/B07NSW9RBQ/ref=sr_1_6?crid=3EDVNS6LIQKI2&keywords=m3+x+8mm+screws&qid=1677476973&sprefix=m3+x+8mm+screw%2Caps%2C93&sr=8-6)
8. [M3 x 10mm Screws](https://www.amazon.com/Socket-Screws-Metric-Stainless-Machine/dp/B0BJ1V4FKY/ref=sr_1_6?crid=3EDVNS6LIQKI2&keywords=m3%2Bx%2B8mm%2Bscrews&qid=1677476973&sprefix=m3%2Bx%2B8mm%2Bscrew%2Caps%2C93&sr=8-6&th=1)
9. [M3 x 35mm Screws](https://www.amazon.com/M3-0-5x35mm-Stainless-Machine-Eastlo-Fastener/dp/B07X2Q33NW/ref=sr_1_19?crid=A5V6SF3S5SII&keywords=m3%2Bx%2B35mm%2Bscrews&qid=1681514731&sprefix=m3%2Bx%2B35mm%2Bscrews%2Caps%2C106&sr=8-19&th=1)
10. [M3 Nylon Locknuts](https://www.amazon.com/SpzcdZa-Stainless-Industrial-Construction-Fasteners/dp/B08LMQC765/ref=sr_1_1_sspa?crid=282O9LSLQ01JO&keywords=m3%2Block%2Bnuts&qid=1681514827&sprefix=m3%2Block%2Bnuts%2Caps%2C93&sr=8-1-spons&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEyTU1XQ09ESVpPWDFLJmVuY3J5cHRlZElkPUEwNzkyMTI5MUxaUlVaT1ZFVFdDRyZlbmNyeXB0ZWRBZElkPUExMDEzOTM5WURaQlZUUzNBV0k0JndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ&th=1)
11. [M4 x 20mm Screws](https://www.amazon.com/Socket-Screws-Metric-Stainless-Machine/dp/B07KRFQJK1/ref=sr_1_6?crid=3EDVNS6LIQKI2&keywords=m3%2Bx%2B8mm%2Bscrews&qid=1677476973&sprefix=m3%2Bx%2B8mm%2Bscrew%2Caps%2C93&sr=8-6&th=1)
12. [M4 x 25mm Screws](https://www.amazon.com/Socket-Screws-Metric-Stainless-Machine/dp/B07KRS36P2/ref=sr_1_6?crid=3EDVNS6LIQKI2&keywords=m3%2Bx%2B8mm%2Bscrews&qid=1677476973&sprefix=m3%2Bx%2B8mm%2Bscrew%2Caps%2C93&sr=8-6&th=1)
13. [M4 Nylon Locknuts](https://www.amazon.com/SpzcdZa-Stainless-Industrial-Construction-Fasteners/dp/B08LMNFS5P/ref=sr_1_1_sspa?keywords=m4%2Blocknuts&qid=1681514944&sr=8-1-spons&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzRU5YVFhHV09QRFJUJmVuY3J5cHRlZElkPUEwMTIzODg3QlQ4SjVKVUcwVDFUJmVuY3J5cHRlZEFkSWQ9QTAzMTgzMDlGQUdMUDM5Vzc4TjImd2lkZ2V0TmFtZT1zcF9hdGYmYWN0aW9uPWNsaWNrUmVkaXJlY3QmZG9Ob3RMb2dDbGljaz10cnVl&th=1)

# Mechanical Configuration

This section details the mechanical design, fabrication and testing of the 3RRS parallel manipulator platform. The 3RRS platform has **3 degrees of freedom and three points of actuation**. Three stepper motors with TMC2209 drives are used for precise actuation. Thanks to the KN-Humanoid club’s 3D printer - 3D printing was used to fabricate most of the structures.

The design was taken from **Aaed Musa**’s Ball Balancing platform [[1]](#footnote-1) with slight modifications. The work is implemented as a demonstration and thus is not suitable for industrial applications.



4

2

3

5

6

1

Figure 3.1 3RRS Ball Balancer mechanical design (description below)

A metallic ball **1** is placed on a resistive touchpad **2** which outputs the position of the ball by detecting the contact point and returning it as a cartesian point. The platform is connected to the actuator legs with a spherical joint **3.** The actuator leg has two links connected by a revolute joint **5.** The lower leg is connected to the rotor of the stepper motor **6** with a revolute joint **4.**

# Electrical Configuration

This chapter outlines the electrical configuration of the project, including the Maker UNO, TMC2209 stepper driver, CNC shield, stepper motor, and resistive touchpad. A NEMA 17 stepper motor provides accurate movement, with manual homing at power-up. A resistive touchpad is employed to track the ball's position, offering a simple and reliable solution.

## Arduino UNO

Maker UNO is a similar microcontroller as Arduino UNO. It has an identical pinout configuration as compared to Arduino UNO. The ease of use, versatile connectivity with ample IO ports and analogue ports, native programming IDE and real time control with low latency make it an ideal choice.

A blue circuit board with many wires

Description automatically generated

Figure 4.1 Arduino UNO pinout diagram. Maker UNO has an identical configuration.

Table 4.1 Maker UNO technical specification.

|  |  |
| --- | --- |
| **Category** | **Specification** |
| Microcontroller | ATmega328P |
| Operating Voltage | 5V |
| Input Voltage (Recommended) | 7V - 12V |
| Input Voltage (Limits) | 6V - 20V |
| Digital I/O Pins | 14 (of which 6 can be used as PWM outputs) |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 40 mA (recommended) |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328P) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Clock Speed | 16 MHz |
| UART | 1 (Universal Asynchronous Receiver/Transmitter) |
| SPI | Yes |
| I2C | Yes |

## TMC2209 V2.0

The TMC2209 is known for its silent operation. Low noise levels, different micro stepping options, current control and low heat generation makes it an ideal choice to control the stepper motors. The is set using a potentiometer screw.

From the stepper documentation,

A black circuit board with many small screws and many small screws

Description automatically generated

Figure 4.2 TMC2209 V2.0 stepper driver pinout diagram.

Table 4.2 TMC2209 Microstepping Configuration.

|  |  |  |
| --- | --- | --- |
| **MS1** | **MS2** | **Microstepping Mode** |
| Low | Low | Full Step (1x) |
| High | Low | Half Step (2x) |
| Low | High | Quarter Step (4x) |
| High | High | Sixteenth Step (16x) |

1/16th step configuration will be used to achieve a smooth motion with minimal jerk.

Table 4.3 TMC2209 technical specifications.

|  |  |
| --- | --- |
| **Specification** | **Value** |
| Operating Voltage (VM) | 4.75V to 29V DC |
| Output Current | 2.0A RMS, 2.4A Peak (per phase) |
| Micro stepping Resolution | Full, Half, Quarter, Eighth, Sixteenth steps |
| PWM Frequency | Up to 28 kHz |
| Control Interface | UART (for advanced configuration) |
| Operating Temperature | -40°C to +150°C |

## CNC Shield

A CNC shield will be used instead of manually connecting the jumpers through a breadboard. The CNC shield has built in decoupling capacitors and integrates with Maker UNO seamlessly. Addition of CNC shield makes electrical management easy.

Micro stepping can be configured by using onboard micro stepping jumpers.

A circuit board with many components

Description automatically generated

Figure 4.3 CNC shield pinout diagram

Table 4.4 Cross-referencing of Arduino pins

|  |  |  |
| --- | --- | --- |
| UNO Pin | CNC Shield Pin | Function |
| Pin 2 | X Step | Step signal for X-axis motor |
| Pin 3 | Y Step | Step signal for Y-axis motor |
| Pin 4 | Z Step | Step signal for Z-axis motor |
| Pin 5 | X Dir | Direction signal for X-axis motor |
| Pin 6 | Y Dir | Direction signal for Y-axis motor |
| Pin 7 | Z Dir | Direction signal for Z-axis motor |
| Pin 8 | Enable | Enable signal for X-axis motor |

In addition to the digital pins, analogue pins are also used. The resistive touchpad is connected to A0, A1, A2 and A3 pins shown in the schematics.

## Stepper Motor

NEMA 17 bi-polar stepper motors from STEPPERONLINE were used for the project. Cost-efficient, precise control and low noise make it an ideal choice. Positional encoders are not used. Instead, homing is done by manually pushing the platform to the bottom point every time it is powered on. When the steppers are powered off the platform automatically falls to the home position.

A black and silver square device

Description automatically generated

Figure 4.4 Nema 17 Bi-polar stepper motor.

A diagram of a wire

Description automatically generated

Figure 4.5 Nema 17 Stepper motor dimensions

Table 4.5 Nema 17 technical specifications

|  |  |
| --- | --- |
| **Property** | **Value** |
| Inductance (mH) | 4.0 |
| Phase Resistance (ohms) | 2.3 |
| Rated Current (A) | 1.5 |
| Step Angle (deg.) | 1.8 |
| Bipolar/Unipolar | Bipolar |
| Holding Torque (Ncm) | 42 |
| Frame Size (mm) | Nema 17 (42 x 42) |
| IP Rating | 40 |

## Resistive touchpad

Traditionally a camera is used to obtain the position of the ball. For this project, a resistive touchpad is used instead. The choice was made due to its simplicity and ease of use.

A **four-wire resistive touchpad** works by detecting touch through two flexible, transparent resistive layers that come into contact when pressure is applied. Each layer has electrodes along its edges: the top layer for the **X-axis** and the bottom layer for the **Y-axis**. To determine the touch point, a voltage is first applied across one layer (e.g., X-axis), and the resulting voltage at the contact point is measured to find the horizontal position. Then, a voltage is applied across the other layer (Y-axis) to measure the vertical position. By alternating these measurements, the touchpad calculates the **X and Y coordinates** of the touch point using the voltage divider principle.

A white rectangular frame with yellow wires

Description automatically generated

Figure 4.6 Four wire resistive touchpad

.

# System Modelling

A PID controller is used to dynamically regulate the platform's orientation by combining three control terms: proportional, integral, and derivative. The proportional term addresses the current error, the integral term eliminates small steady-state errors, and the derivative term predicts and counters rapid changes in the error for improved stability. This chapter outlines the implementation and modelling of the PID control system, explaining how these terms work together to optimize performance. It also covers the tuning of the controller parameters for optimal response, stability, and accuracy in controlling the platform's orientation under various conditions.

## PID Controller

A proportional–integral–derivative controller (PID controller or three-term controller) is a feedback-based control loop mechanism commonly used to manage machines and processes that require continuous control and automatic adjustment. It is typically used in industrial control systems and various other applications where constant control through modulation is necessary without human intervention. The PID controller automatically compares the desired target value (setpoint or SP) with the actual value of the system (process variable or PV). The difference between these two values is called the error value [1].

A screenshot of a computer

Description automatically generated

Figure 5.1 A block diagram of a PID controller in a feedback loop. r(t) is the desired process variable (PV) or setpoint (SP), and y(t) is the measured PV. [1]

The overall control function is described by,

|  |  |  |
| --- | --- | --- |
|  |  | (5.1) |

In the case of a ball-balancing platform as described earlier, the proportional gain responds to the distance between the ball's current position and the set point. The derivative controller accounts for the speed at which the ball is moving away from the set point, while the integral term addresses any small steady-state error.

### Proportional Control

The purpose of the proportional controller is to minimize the steady-state error by increasing the system's proportional gain. By introducing a constant , known as the proportional gain, the proportional response is adjusted such that the steady-state error decreases inversely with the proportional gain. The proportional term, or output signal, is defined by equation (5.2).

|  |  |
| --- | --- |
|  | (5.2) |
|  |  |

However, if the gain is set too high, it may lead to instability in the system by causing excessive output fluctuations. On the other hand, if the gain is too low, the system may struggle to effectively handle both external and internal disturbances.

### Integral Control

The integral controller is proportional to both the duration and magnitude of the error over time. It corrects any accumulated offset in the system by summing the errors over a finite period, effectively eliminating steady-state error. However, the integral gain can sometimes degrade the system's response, potentially causing transient or oscillatory behaviour. The output signal from the integral term is given by equation (5.3).

|  |  |
| --- | --- |
|  | (5.3) |

### Derivative Control

The derivative controller is determined through the calculation of the error’s response slope over time. The slope of the error is then multiplied with , the derivative gain. The derivative term is defined in equation (5.4).

|  |  |
| --- | --- |
|  | (5.4) |

## Inverse Kinematics of the system

In this sub-chapter, the inverse kinematics equations required are derived to orient the platform in such a way that it sends the ball towards setpoint. The platform is supported by three legs, each modelled as a two-link manipulator connecting the base to the platform. The configuration of each leg consists of two revolute joints and two rigid links, forming a planar structure.

Given a unit normal vector of the platform plane, the IK equations then compute the required angles of the revolute joint [2].

A diagram of a hexagon with points and lines with Great Pyramid of Giza in the background

Description automatically generated

Figure 5.2 Kinematic Diagram of the ball balancing platform [2]

Table 5.1 Description of symbols in Figure 4.2

|  |  |
| --- | --- |
| SYMBOL | DESCRIPTION |
| O | Origin of the global co-ordinate system |
|  | Revolute Joints |
|  | Spherical Joints |
| P | Platform Plane origin |
| Z’ | Unit normal vector of the platform plane |

The system is divided into three separate kinematic chains. The origin for the global frame (O) is marked at the mid-point of the base plane. The mid-point of the platform plane is represented by .The required rotor angles is calculated as a function of tilt of the platform plane represented by and the height between the platforms represented by h.

|  |  |
| --- | --- |
|  | (5.5) |

A diagram of a triangle with lines and points

Description automatically generated

Figure 5.3 Kinematic diagram for a single chain

Figure 4.2.2 represents one kinematic chain out of three. While the other chains are very similar, derivations will be performed individually. The derivation of the first chain is as follows.

A diagram of a circle with lines and circles

Description automatically generated

Figure 5.4 Kinematics of Link 1 and Link 2

**Unit normal Vector:**

**Equations for :**

|  |  |
| --- | --- |
|  | (5.6) |

**Equations for :**

|  |  |
| --- | --- |
|  | (5.7) |

**Equations for :**

|  |  |
| --- | --- |
|  | (5.8) |

## Equations of Motion for the ball

To simulate the movement of the ball, a model describing the equations of motion of the ball is necessary. The three-dimensional ball-on-platform system will be split into two separate two-dimensional ball-on-beam systems, as shown in Fig. 5.5. These systems will share the same equations but with different variables. Index 1 will refer to the system observed in the x-z plane of the room, while index 2 will correspond to the system viewed in the y-z plane. The equations will be derived for the first system and later translated into their equivalent form for the second system.

A diagram of a rectangular object with a circle and a circle with a circle

Description automatically generated with medium confidence

Figure 5.5 Three-dimensional system represented as a pair of two-dimensional system [3]

A diagram of a triangle and a circle

Description automatically generated

Figure 5.6 Free body diagram of the ball in two dimension [3]

The goal is to model the motion of the ball in XY plane given the tilt angles of the XY plane. The equations of the model are [3].

**Lagrangian Method: Equations of Motion**

The Lagrangian Method derives the equation of motion through the relation of kinetic and potential energy of the system. This method is particularly useful for complex systems with multiple degrees of freedom. Below are the detailed derivations and descriptions.

**Lagrangian Equation:**

L (Lagrangian) is the difference between the kinetic and potential energy of the system:

**Kinetic Energy:**

Where:

Thus, the total kinetic energy becomes:

Potential Energy:

**Lagrangian:**

**Equation of Motion (x-direction):**

Substituting these into Equation 4.1, the differential equation becomes:

Simplifying for , the equation of motion is derived as:

|  |  |
| --- | --- |
|  | (5.9) |

**Equation of Motion (y-direction):**

Applying the same method to the y-direction, the equation of motion is:

|  |  |
| --- | --- |
|  | (5.10) |

# Constructing the Ball Balancer

# References

|  |  |
| --- | --- |
| [1] | Wikipedia, "PID Controller," [Online]. Available: https://en.wikipedia.org/wiki/Proportional–integral–derivative\_controller. |
| [2] | B. Zi, C. Liu and Y. Yueqing, "Dynamics of 3-DOF spatial parallel manipulator with flexible links," 2010. |
| [3] | M. T. ALEXANDER HASP FRANK, "Construction and theoretical study of ball balancing platform," Stockholm, Sweden, 2019. |
| [4] | N. Apazidis, Mekanik II - Partikelsystem, stel kropp och analytisk mekanik, Kartonnage, Svenska, 2019. |

1. https://www.youtube.com/watch?v=v4F-cGDGiEw [↑](#footnote-ref-1)