***Design and Control of a Ball-Balancing Table***



**Under Supervision:**

Dr. Abdelrahman Ahmed Morsi

Dr. Abdelfattah Mahmoud Mohamed

**Project Group:**

* Kerolos Emad
* Logina Mamdouh
* Maryam Mostafa
* Menna Mohamed
* Michael Emad
* Nermeen Mohamed
* Steven Hany
* Youstina Kamal

Contents:

**ABSTRACT**……………………………………………………………………….……………………………………………………… [4]

**INTRODUCTION**

(1-1)General motivation

(1-2) application

(1-3)Table (1) quantities and its units

1 Components of Ball Balancing Table ................................................................................................4

1.1 RC Servomotors ....................................................................................................................... 4

1.1.1 Overview..............................................................................................................................4

1.1.2 Theory of Operation.............................................................................................................6

1.5 Mechanics of Ball Balancing Table............................................................................................ 9

2 Pulse Width Modulation and RC Servomotor.................................................................................10

2.1 Understanding Pulse Width Modulation (PWM) ....................................................................10

2.2 Driving an RC Servomotor with PWM.....................................................................................11

4 System Modeling ...........................................................................................................................15

4.1 Non-Linear Equation of Motion..............................................................................................17

4.1.1 Lagrangian Method............................................................................................................17

4.1.2 Newton’s Law of Motion....................................................................................................19

4.2 Modeling of Ball Balancing Table System ...............................................................................20

4.2.1 Linearization around Operating Point ................................................................................22

4.2.2 Obtaining Transfer Function of Plant..................................................................................22

4.3 Modeling of Actuator.............................................................................................................23

4.4 Cascade Control of Ball Balancing Table.................................................................................23

5 Performance Measures..................................................................................................................25

5.1 Understanding Percentage Overshoot, Settling Time, Peak Time and Steady State Error.......25

5.1.1 Damping Ratio (𝜉) ..............................................................................................................25

5.1.2 Natural Frequency (𝜔n) ......................................................................................................25

5.1.3 Percentage Overshoot .......................................................................................................25

5.1.4 Peak Time (tp).....................................................................................................................27

5.1.5 Settling Time (ts).................................................................................................................27

5.1.6 Steady State Error ..............................................................................................................27

6 Control System Design ...................................................................................................................28

6.7 PID Controller ........................................................................................................................45

6.7.1 Sample Design with PID controller .....................................................................................46

6.7.2 In-Lab Exercises..................................................................................................................48

**References** ............................................................................................................................................56

**ABSTRACT:**

This project centers around the application of PID theory and how this theory can be applied to model and predict the movement of a ball on a flat surface. The PID system provides feedback and response to the system in order to allow the control system to automatically adjust itself with the goal being to balance a ball on a table and always have the ball return to the center. This is accomplished by gathering the distance from an Camera and using PID theory to predict the appropriate K, Kp, and Kd values for the system to correctly adjust itself. After attempting to predict the values using PID theory, the best results were instead obtained using trial and error. Through trial and error, the team was able to successfully identify the appropriate values and build a PID control system that balanced a ball on a table.

**Introduction:**

For our final year of study we were to undertake a project in electrical lasting year. The project chosen was the ‘design and control of Ball Balancing table project. We were to design a system whereby we could balance a ball on a flat surface using actuators. This project was a great opportunity to gain knowledge in the field of Control System. The goal of the project is to build and implement a device that can balance a ball at any set point by tilting the flat surface allowing the ball to roll. The surface is tilted by actuators located below it. The mechanics by which the actuators transfer torque to the flat surface will have to be designed and built. The control system will have to drive the motors to make the ball move to the allocated position. In addition a feedback system will need to be implemented so the balls current position can be known at all times.

**THEORY:**

The purpose of this study was to demonstrate how feedback influences the response of a control system by constructing a PID controller. A control system is a device that manages and regulates the behavior of a particular system. The system in our case is the ball balance table. In this system, a table must be able to balance a ball and return the ball to the center of the table if moved. There are two different types of systems, open loop and closed loop. Open loop systems have no feedback and are simply based on the input while closed loop systems use feedback to improve the control of the system. The principle being tested by this experiment is how feedback can improve the system. This is achieved by the construction of a PID controller. The PID controller is a type of feedback controller where P stands for proportional, I for integral, and D for derivative. The “P” term is responsible for producing an output value that is proportional to the current error that the system is calculating. The “I” term is essentially the sum of all the instantaneous error and is responsible for eliminating the steady state error that the P controller would produce. The “D” term is responsible for predicting the future system behavior and using the predicted behavior to improve the time and stability of the system.

When all three parts are put together in one controller, the system should be able to achieve a steady state of oscillation. The output of the system is reliant of each part of the PID controller, and each term has a corresponding gain that accompanies it. The “P” produces a proportional gain and this value has a direct result on response speed and percent overshoot. The “I” produces an integral gain, which when combined with the proportional gain can produce an increase in rise time and settling of the system. The derivative gain that is produced by the “D” term will help to decrease the overshoot value and settling time.

**SOFTWARE DESIGN**

The software for the ball balancing robot, found in the Appendix, was compiled Arduino microcontroller. The software allows a user to set the zero points for the Servo motor, change tuning parameters, and watch as the robot tries to balance the ball on the table with those parameters. If the user is not satisfied with the results, a reset button could be used, and the robot would stop and let the user change their tuning parameters. These features for the robot were accomplished using two libraries which included the Arduino PID library and the Servo library. The software started with several defining statements for the values of some of the pins used by the electronic components. There was also a structure declaration which stored several key values used throughout the program. Some of the functions include in the program included a function to display the current tuning parameters and mode, a function to receive a distance reading from the Camera, functions to control the servomotor, a function to control the modes, and a PID mode function for when the controller is started. The main function of the program consisted of several calls to Pin Mode and digital Write to set the state of some of the pins.. A declaration for the structure mentioned earlier was made, and this structure was passed by reference to an infinite for loop running the control function.

Components of Ball Balancing Table

The main parts of “Ball Balancing Table” can be seen below.

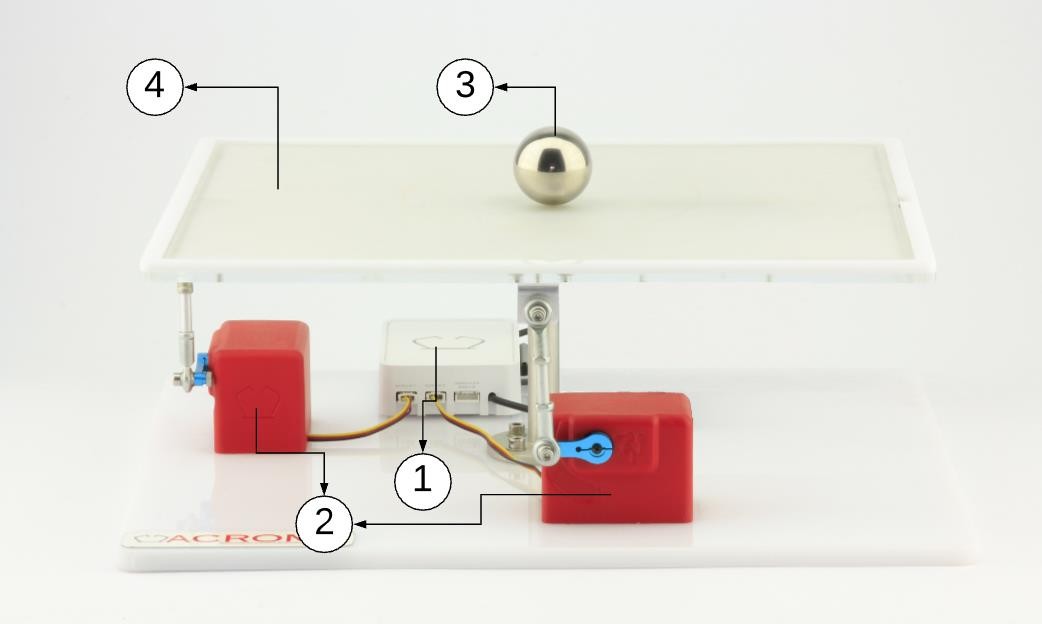


Figure 1.1: Components: 1. Power Distribution Box

2. RC Servos 3. Ball 4.Camera and Raspberry Pi

RC Servomotors

Overview

RC servos are electromechanical devices that convert electrical signals to rotary movement. They provide simple and handy solutions to most of control and robotic applications.



Figure 1.2: RC Servomotor

RC servomotors consist of several parts that include:

Controller: Circuit that is responsible of reading the control signals (PWM signals –more about PWM signals can be found in Chapter 2) and controlling the motor accordingly. Controller circuits determine the type of servos; they can be either digital or analog. Analog servos handle PWM signals up to 50 Hz frequency. Digital servos handle these PWM signals more precisely. They can decode signals up to 330 Hz. This difference in decoding can provide more torque to the motors. In addition to that; digital servos can be programmed to change direction, dead band width etc. Generally; digital servos are more advantageous than analog servos in expense of cost and power consumption.

Potentiometer: Position feedback of the main shaft is provided by the potentiometer. It is attached to the driveshaft so the rotation of the driveshaft results in different resistances on the potentiometer. By reading the resistance values, controller knows exact driveshaft angle.

Motor: The motors of the servos are commonly high speed DC motors which are controlled by H-bridges located inside their controller circuitry.

Gearbox: The gear set is located between the driveshaft and the motor. It regulates the RPM of the motor resulting in lower movement speed and more torque.

Driveshaft: Driveshaft is the output of the whole system. It is the actual component that rotates to the desired angle.

Connector: They generally have three pins that carry “+”, “-“ and “signal” to the controller. Connectors may have different color codes depending on the manufacturer.

A small electric motor with wires and gears

Description automatically generated

Figure 1.3: RC Servo Components

* + 1. Theory of Operation

The servo is actually a closed loop control system that requires continuous input signal. Servo operation can be explained with few basic steps:

* + - 1. Controller decodes the PWM input signal and converts it into a voltage that corresponds to an angle.
      2. Controller reads the potentiometer voltage values and determines the shaft position.
      3. Controller calculates the error from the difference between input and potentiometer voltage.
      4. Controller converts error into the H-bridge output.

A diagram of a computer

Description automatically generated

Figure 1.4: RC Servo Theory of Operation

**System Modeling**

Figure 1 shows how the Ball Balancing Table is put together. It describes how the motors (motor x and motor y) are firmly attached to the base plate. These motors are connected to the table with joints that allow movement in all directions. The table itself is connected to the base plate through a joint that lets it move in two different ways.

The model also explains the specific connections and measurements of the table that are important for understanding how the whole system works:

A diagram of a diagram

Description automatically generated

Figure 1: The Physical Model of the Ball Balancing Table

|  |  |  |  |
| --- | --- | --- | --- |
| Symbols Definition Value Unit | | | |
| Lx | Beam Length in x-direction | 0.2 | [m] |
| LY | Beam Length in y-direction | 0.2 | [m] |
| rm | Motor Arm Length | 0.02 | [m] |
| rb | Radius of the Ball | 0.02 | [m] |
| mb | Mass of the Ball | 0.024 | [kg] |
| Jb | Rotational Inertia of the Ball | 0.0000096 | [kg\*m2] |
| G | Acceleration due to Gravity | 9.81 | [m/s2] |
| α | Plate Angle about x-axis | (Variable) | [degrees] |
| β | Plate Angle about y-axis | (Variable) | [degrees] |
| θx | Motor Angle for x-axis | (Variable) | [degrees] |
| θy | Motor Angle for y-axis | (Variable) | [degrees] |

table 1

Before deriving the system model, it is essential to consider the following assumptions:

- The model assumes that the contact between the ball and the plate is maintained under all circumstances.

- It is assumed that the ball rolls on the table without slipping.

- To simplify the model, all friction forces and resulting torques are neglected.

**Lagrangian Method:**  
The Lagrangian Method derives the equation of motion by considering the relationship between the kinetic and potential energies of the system. It is particularly useful for systems with more than one degree of freedom. The Lagrangian Equation is formulated as:

In the context of the ball balancing table system, the Lagrangian method will be used to derive the equations of motion, considering the kinetic and potential energies of the ball and the table.

A diagram of a ball and a ball

Description automatically generated

Figure 2: Free Body Diagram

To derive the equation of motion, we first need to obtain the kinetic and potential energies of the system. The total kinetic energy of a rolling ball (assuming no slipping) can be described as follows:

The kinetic energy of the ball consists of two components: translational and rotational kinetic energies.

1. **Translational Kinetic Energy**
2. **Rotational Kinetic Energy**

Next, we will derive the potential energy of the system, which involves the gravitational potential energy of the ball and any potential energy associated with the tilting of the table.

Therefore, the Lagrangian (L) for the ball balancing table system, considering the kinetic and potential energies derived earlier, is given by:

The same for y

Using relationship between motor angles and plate angles:

A diagram of a ball and a ball

Description automatically generated

Figure 3: Relationship between Motor Angles and Plate Angles

To obtain the transfer function of the system, the differential equations must be linearized about the operating point (*x=0, y=0*). For small angles:

By using the value of system parameters which were listed in Table 1

# DC Motor Control Section

**Abbreviation:**

As we talk previously about using servo motors to control the ball balancing table we thought to make it a little bit complex and to make it with dc motors too to enrich our knowledgement and gain more experience. it takes the same steps as the first project which mentioned previously except that when the readings of the axis reaches the arduino uno and converted to angles, it then send the angles to 2 arduino nanos which will send separately for each motor the angle of the axis it is responsible for.

**DC Motors and Control of DC Motors**

**Overview:** DC motors are electrical machines that convert direct current electrical power into mechanical power. They are widely used due to their simplicity, ease of control, and versatility.

**Types of DC Motors:**

1. **Brushed DC Motors:**
   * Have brushes and commutators.
   * Simple design and easy to control.
   * Suitable for applications requiring variable speed and torque.
2. **Brushless DC Motors (BLDC):**
   * Do not have brushes, reducing maintenance.
   * Higher efficiency and longer lifespan.
   * Common in applications where high reliability is crucial, such as in aerospace and medical devices.

**Control of DC Motors:** DC motor control can be achieved using various methods, each offering different levels of precision and performance. The primary methods include:

1. **PWM (Pulse Width Modulation):**
   * Controls motor speed by varying the duty cycle of the voltage applied to the motor.
   * Provides precise control and is efficient in terms of power consumption.
2. **H-Bridge Circuit:**
   * Allows the motor to be driven in both directions by switching the polarity of the voltage applied.
   * Consists of four switches (transistors or MOSFETs) arranged in an H-shaped configuration.
3. **PID Control:**
   * A closed-loop control system that uses proportional, integral, and derivative components to maintain the desired motor speed or position.
   * Commonly used in applications requiring precise control, such as robotics and industrial automation.

**Applications of DC Motors:** DC motors are used in various applications, including:

* Electric vehicles
* Robotics
* Industrial machinery
* Household appliances

**Conclusion:** DC motors play a critical role in many modern applications due to their controllability and efficiency. Understanding their operation and control mechanisms is essential for designing effective motor-driven systems

**Components:**

The control system for the Ball Balancing Table involves several key components:

 **Raspberry Pi**: Captures the ball's position.

 **Arduino Uno**: Implements PID control to determine the necessary angles for balancing.

 **DC Motors with Encoders**: Motors equipped with encoders that generate pulses corresponding to the motor shaft's rotation..

 **Arduino Nano**: Each motor is connected to an Arduino Nano, which processes the angle data received from the Arduino Uno.

 **Motor Driver**: Typically, a motor driver L298N is used to interface the Arduino Nanos with the DC motors, providing the necessary current and voltage for motor operation.

 **Power Supply**: An external power supply is used to provide adequate power to the motors and the motor drivers..

**Control Workflow**

**Step 1:**

The Raspberry Pi, equipped with a camera ,continuously monitors the ball's position on the table. This position data is processed and transmitted to the Arduino Uno via Uart serial communication.

**Step 2:**

The Arduino Uno, after processing the ball position data through the PID algorithm, sends the calculated tilt angles for the X and Y axes to the respective Arduino Nanos. done via i2c serial communication.

**Step 3:**

The encoder attached to each motor generates pulses as the motor shaft rotates. These pulses are read by the Arduino Nano, which counts the number of pulses to determine the motor's position and speed.

The Arduino Nano calculates the motor's position by counting the encoder pulses and determines the speed by measuring the rate of pulse generation. The position is typically calculated in terms of encoder counts, which converted to angular position The Arduino Nanos received as input .

**Step 4:**

The position and speed information from the encoders is used as feedback for the PID control system. This feedback allows the control system to:

* **Adjust for Errors**: Correct any deviations from the desired position by adjusting the motor speed and direction.
* **Improve Stability**: Reduce oscillations and overshoot by providing real-time feedback for more precise control.
* **Enhance Responsiveness**: Quickly respond to changes in the ball's position, ensuring the table remains balanced.

### Conclusion

The DC motor control in the Ball Balancing Table project is a critical component that translates the calculated angles from the PID controller into precise mechanical adjustments. By utilizing Arduino Nanos, motor drivers, and DC motors, the system achieves responsive and accurate control, ensuring the ball remains balanced on the table. This segment of the project showcases the importance of motor control in real-time systems and highlights the effective use of microcontrollers to achieve complex tasks.