ME2400

Measurement Instrumentation and Control
PROJECT REPORT

FINAL DUE DATE: 26-04-2019

GROUP NO: 10

GROUP MEMBERS

1) D.NAVEEN REDDY ME17B140
2) Y.SRIHARI RAO ME17B118
3) VARUN SANKLECHA ME17B073
4) S. TARUN PRASAD ME17B114

Table of Contents

Abstract

Recognising the challenges involved in the problem statement, we have come up with strategies to:
1)Design the mechanical model and identify the manufacturing techniques required.
2)Instrument the system with a required sensor to track the ball movement.
3)Identify the actuator required to achieve the task, taking torque, angular speed, etc into account.
4)Select the microcontroller in order to control the motor properties.
5)Identify suitable signal conditioning for the sensor outputs.
6)Calibration of the sensor and filtering the data.

We have implemented PID control algorithms in order to design the system required for achieving the problem statement.

Flow Chart

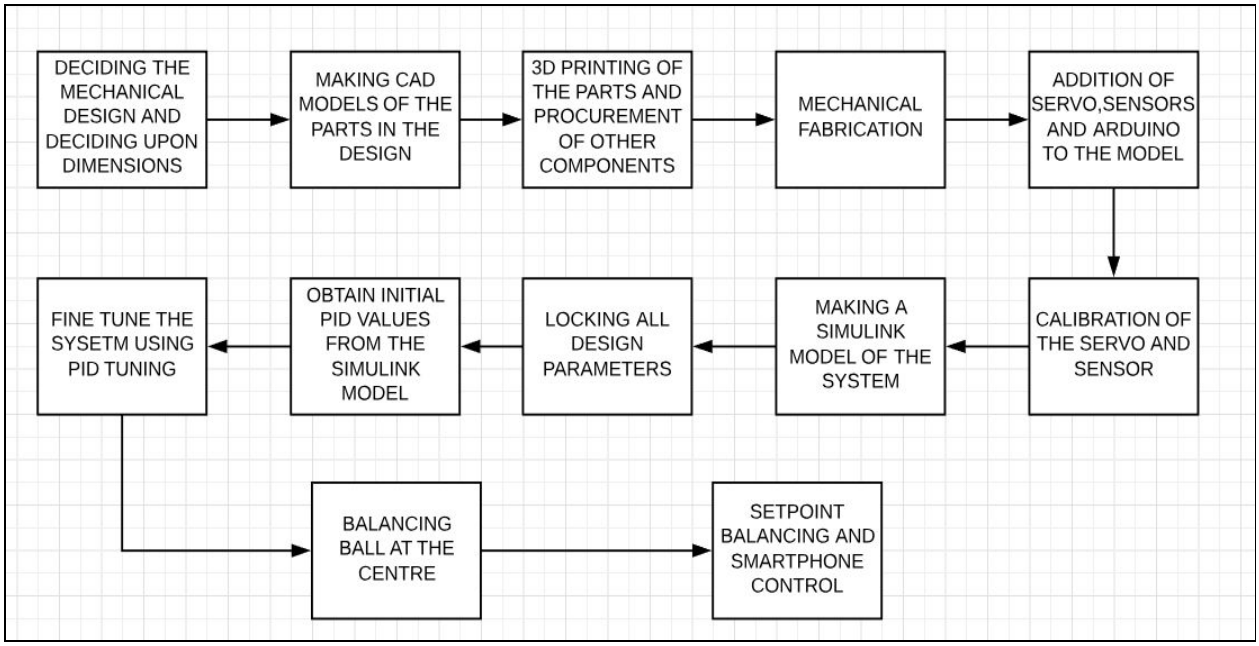

Fig. 1 Flow chart of of the entire project

CAD Model Design

Components bought:

- Plywood
- PLA Filament
- M4 Bolts
- M4 Nuts
- M3 Bolt
- M3 Nut

Components Manufactured:

- The middle groove track slot
- Groove track with ultrasound sensor slot
- Stand with mounting for servo
- Coupling between servo and groove track
- Secondary stand for support

We had fabricated our model using 3D-printing due to the accuracy it offers. It allowed us to manufacture :

- A perfect groove for the selected ball.
- An accurate sector angle for the groove which doesn't interfere with the sensor range.
- Perfect levelled mounting of the sensor.

We selected a track length of 43 cm by optimising the tradeoff between the torque which the selected servo offers and the maximum length we can go to, also taking into account the manufacturing limitations. A horizontal mounting of the sensor was found optimum and hence the track width was selected to accommodate this type of mounting. The ball size was decided by analysing the sensor's performance on different balls and diameters of 35 and above was found to be sufficient. We zeroed in on the golf ball which weighs around 50 g and has a diameter of 43 cm.

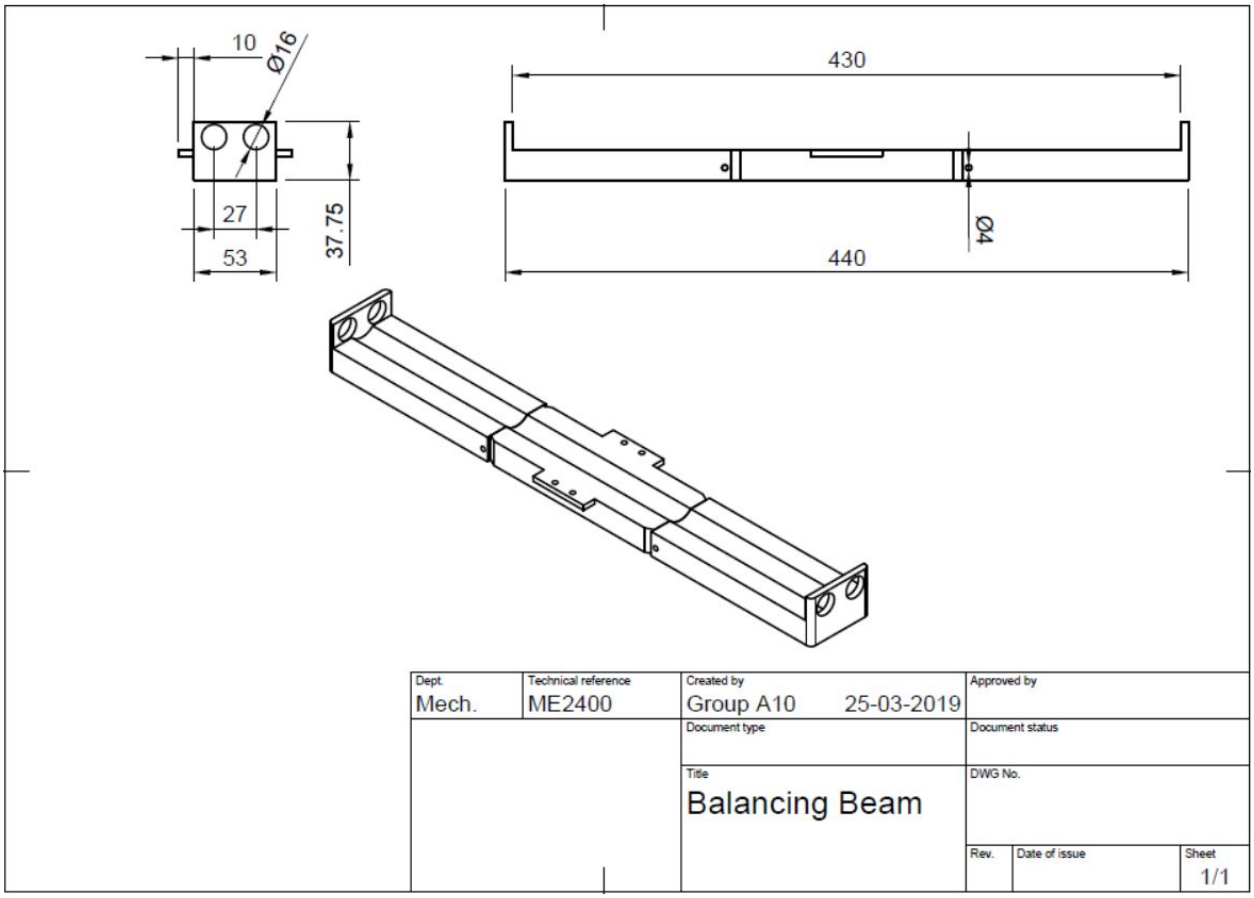

Fig. 2 CAD model for the groove

Electronic Circuit Diagram

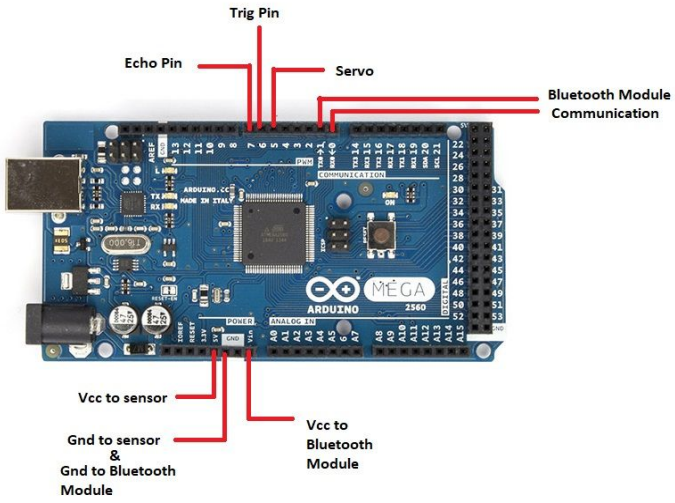

Fig. 3 : Electronic circuit diagram.

Justification: We tried designing the simplest possible circuit with all the necessary provisions for an effortless electronic debugging. Also not compromising on the amperage demanded by the components the connections were given.

Sensor:

HC SR 04 ULTRASOUND SENSOR

Specifications needed: A non-contact sensor which can measure distance of ball up to 80 cm. should work under most of the ambient conditions. Should work with 5 V or 3.3 V so that it can be directly connected to Arduino.

Specification of sensor:

Working voltage: 5V

- Working current: 15 mA
- Working frequency : 40 Hz
- Maximum range: 4m
- Minimum range: 2cm
- Measuring angle: 15 degree
- Trigger input signal: 10uS TTL pulse
- Echo output signal: Input TTL lever signal and the range in proportion
- Dimension: 45*20*15mm

Non-contact type of sensor is better than contact type of sensor. So, in non-contact type two sensors are mostly used to measure distance. They are Ultrasound sensor and IR sensor. Ultrasound sensor has many advantages above the IR sensor. Ultrasound sensor has good reproducibility in a variety of ambient conditions. Whereas the IR sensor will drift its values depending on ambient light conditions.

Fig. 4 Ultrasonic sensor

Correction Element

Servo Motor

For performing PID on the error in controlling ball position and velocity we wanted precision in angle control and hence have decided to use a Servo Motor. Calculating torque and angular velocity requirements we have decided to use the **Tower Pro MG995** motor.

Its rated torque and rpm characteristics are as follows:

Torque :	**4.8V:** 130.54 oz-in (9.40 kg-cm) **6.0V:** 152.76 oz-in (11.00 kg-cm)
Speed:	**4.8V:** 0.20 sec/60° **6.0V:** 0.16 sec/60°

Fig. 5 Servo motor and its specifications.

https://servodatabase.com/servo/towerpro/mg995.

Microcontroller:

Arduino:

Arduino gives an extremely flexible and easy to program microcontroller that suited very well for our level of understanding and also compatible PID tuning. As we individually had some experience with Arduino, we finally decided to go along with it. Mega 2560 model was decided as it was cheap and had more than enough ports for all our needs.

Sensor model number: HC SRO4 ULTRASOUND SENSOR

Fabrication of the CAD model and assembly of sensor, actuator and controller.

The Structure:

The model was 3D printed. This decision was made to reduce manufacturing time and to give flexibility for design. This also gave us high accuracy between CAD design and an actual model. The structure was made in three separately 3D printed parts and was joined together with the help of fasteners.

The different parts were manufactured separately and designed to fit together in perfect harmony.

Provisions were made for Ultrasonic sensor and servo motor.

The earlier design was modified due to its interference with the proximity sensor. Some filing, drilling was necessary to fit in the various parts into each other. The final design was fixed making sure the model was completely ready for tuning.

FLOW CHART

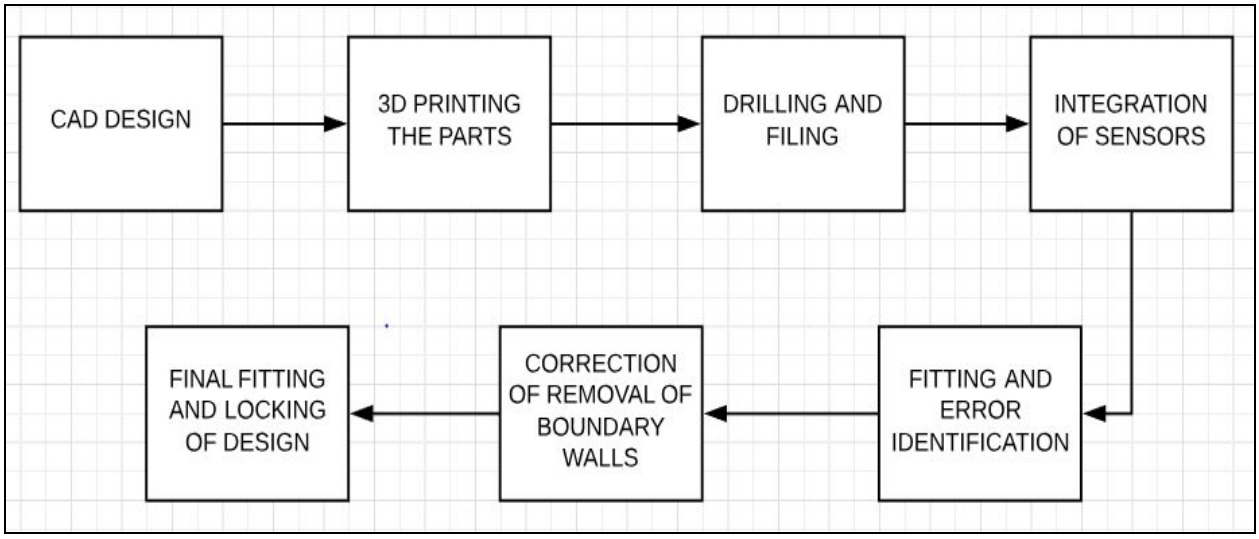

Fig. 6 Flowchart for manufacturing process

Arduino Program

Link to the Code:
https://drive.google.com/drive/folders/1MVu8q8k4G_5XHI3BqD-89nSKmITto7ZR?usp=sharing

Connection diagram of sensors and actuators to arduino

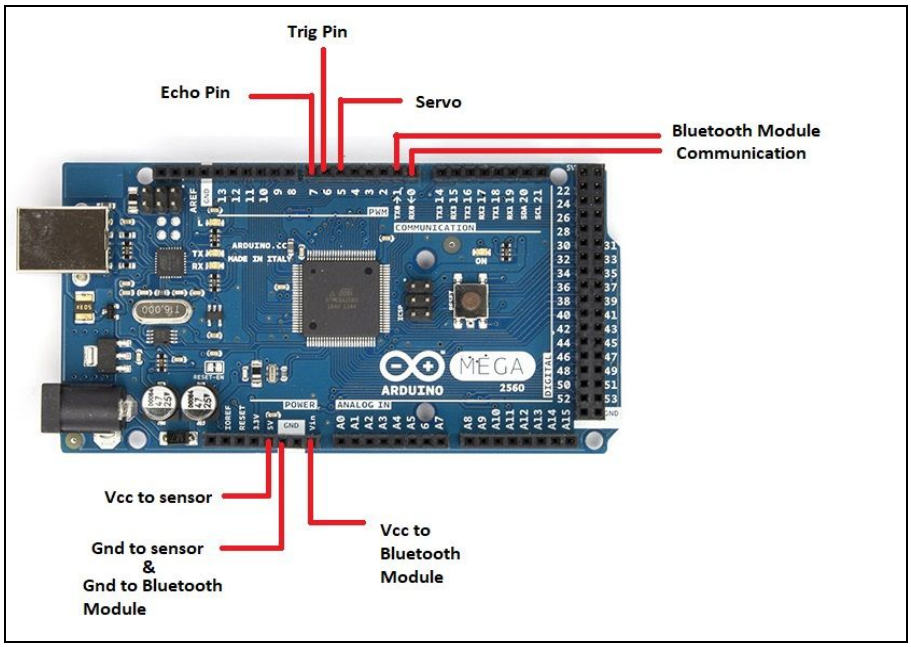

Fig. 7 : Circuit diagram

Signal conditioning :

The sound sensor doesn't give proper value always. It has some noise and also gives spikes in the values. To overcome these issues the arduino code is written in such a way that whenever there are spikes in the value from the sensor those values will be ignored. For example when the sound sensor gives value more than maximum length of the track that input value will be ignored for computation.

Controller Design :

PID control is employed to achieve the task of balancing the ball at the setpoint. Proportional control is the most fundamental control to employ.in addition to that to reduce the amplitude of oscillations integral control is employed. To ensure smoothness in control having in a predictive way differential control is employed. At first the Kp value from the Matlab-Simulink is taken to start the tuning. Gradually the Kp value is increased until the oscillations were good and were slowly decreasing. Then Ki value obtained from the Simulink is used to start Integral tuning. The control was having lag with only PI tuning. So Kd value is increased and the control started becoming smooth and time taken for settling at setpoint reduced drastically. Finally we finalized on the Kp, Kd ,Ki values mentioned below

Kp = 0.6 Ki= 0.15 Kd = 0.38

```
float Kp = 0.6;                                          //Initial Proportional Gain
float Ki = 0.15;                                          //Initial Integral Gain
float Kd = 0.38;                                         //Intitial Derivative Gain
double Setpoint, Input, Output, ServoOutput;
int i,j,s=21.5;
```

Fig. 8 Screenshot of the final tuned PID values

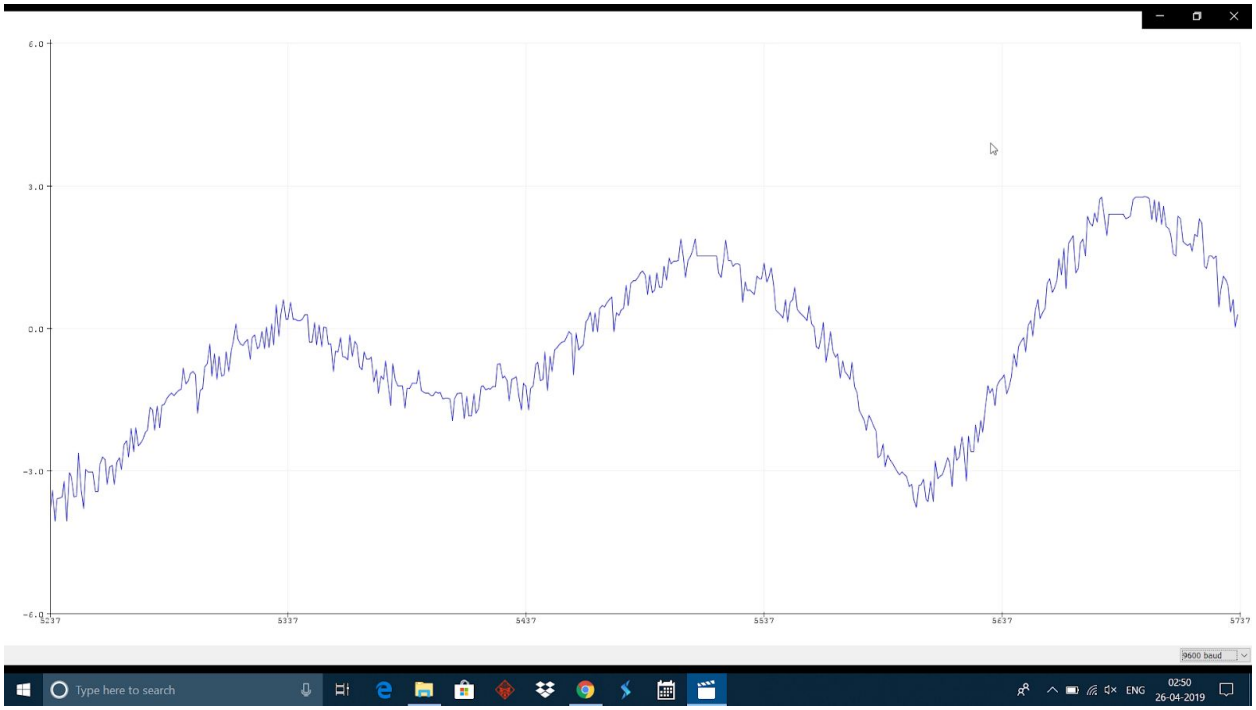

Fig 9: Error vs time recorded in the serial monitor

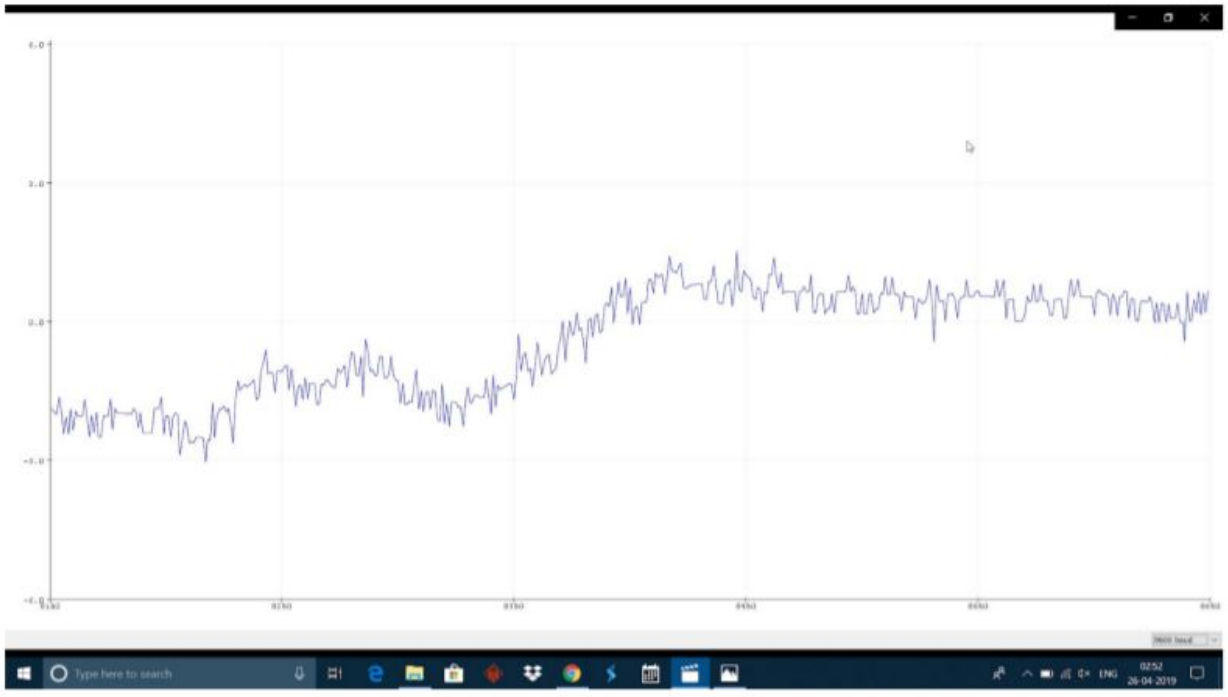

Fig 10: Error vs time recorded in the serial monitor

Arduino Model & Specifications : Arduino Mega 2560

The **Arduino Mega 2560** is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs

(hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

Arduino programme algorithm :

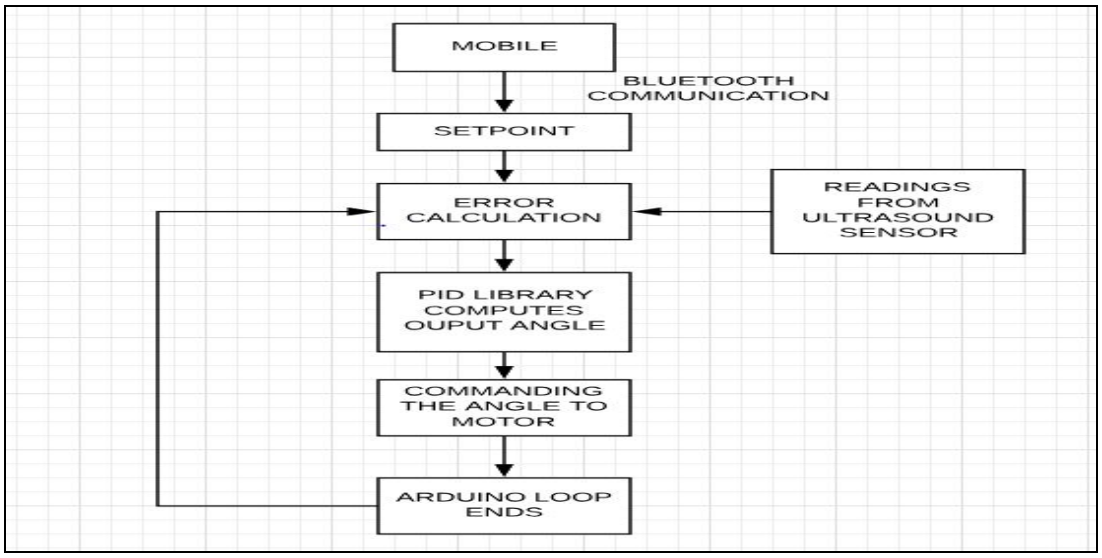

Fig 11: Arduino Algorithm flowchart

Simulink Model

The Simulink model is a standard PID control block diagram with the beam's inclination as the input and the ball position on the beam as the output. The acceleration equation on the ball was written and Laplace Transform was applied to obtain the necessary transfer function. The tuning interface in Simulink was used to obtain the initial PID values.

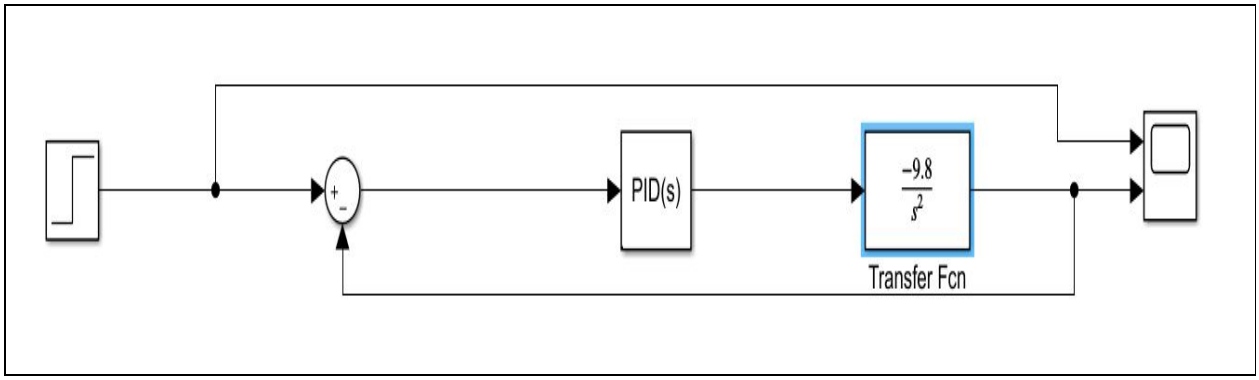

Fig. 12 Simulink Block-Diagram

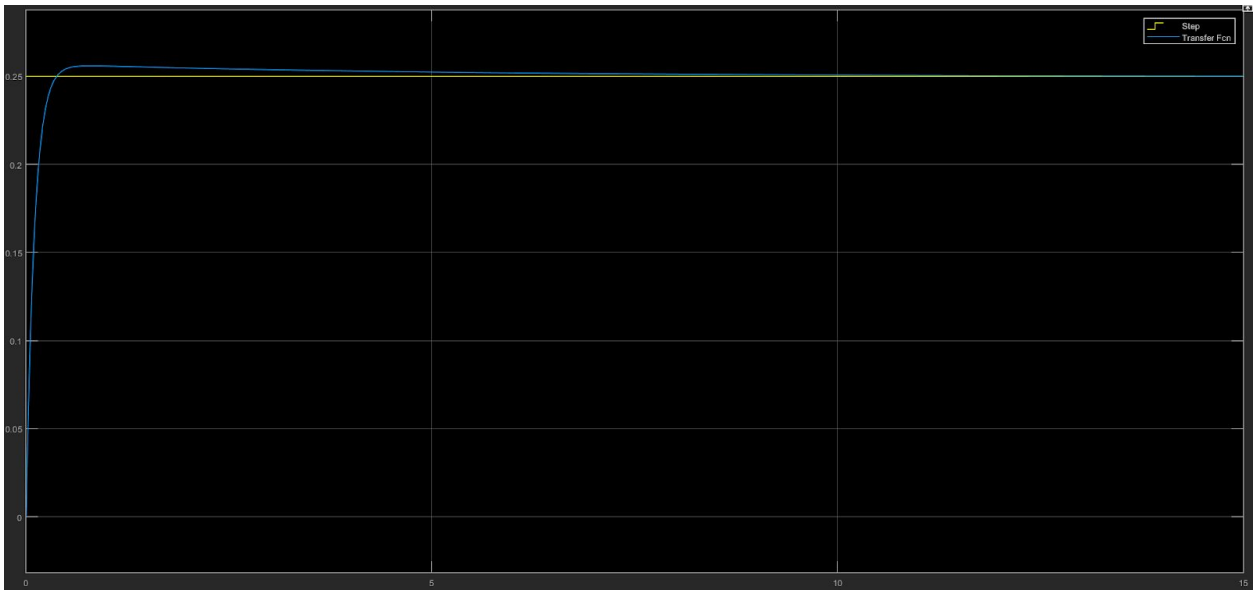

Fig. 13 Process curve

Integration of Simulink model and original model.

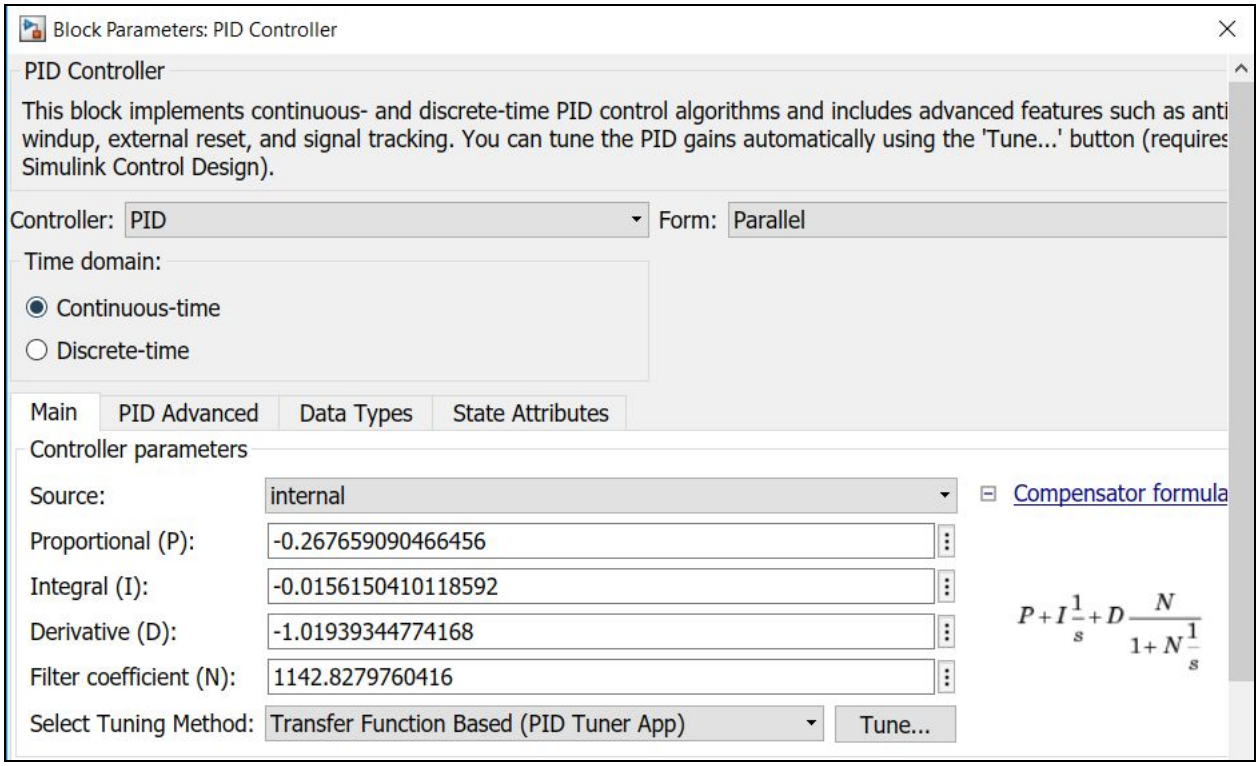

Fig. 14 Tuned Values

The tuning interface in Simulink was used to obtain the initial PID values. These values were used as the initial values for tuning and we started tuning the actual model further for the deviations it showed in balancing for the Simulink predicted values.

Work done for additional marks

- Increase in length of groove track: **43 cm** instead of the minimum specified length of 15 cm.
- Setpoint balancing: the ball can be balanced at any point on the track.
- Fixing the set point from any regular smart-phone: Done through a Bluetooth module.

Challenges faced and how we overcome it as a team.

The first challenge was we faced was the division of task. Since this project is also supposed to be a learning experience we decided to have every group member involved in every aspect of the project with someone specializing. This made sure that if a problem occurred and the one person who was responsible wasn't present we could still solve it.
The next problem we came across was the finalising the design. While some members of the group had a four-bar mechanism in mind others wanted a single bar connected to the servo in the centre. We went through the pros and cons of both and realised they each had their merits and demerits. In the end, a single bar mechanism seemed simpler for tuning as it was mathematically-less complex and thus we moved forward with that design.
We also faced a lot of issues with electrical and tuning aspect but by that time we had reached a certain level of synchrony between us and sorted out those issues much faster.

The design and tuning were the phases that took the most amount of time. Although writing the Arduino code was a simple task, the tuning part was especially difficult as the values changed with every parameter. The manufacturing part was relatively easier as we went for 3D printing. The design took time as we needed to finalise a design that everyone agreed upon. The Simulink model although challenging was designed keeping some approximations in mind.

We believe we have a come a long way as a team this semester owing to this project and course. We learnt not only on the technical aspect but also on how to manage time and work with each other and to trust one another on their work.

From a technical viewpoint, we learnt about PID and its applications. We learnt how to design systems that were practical and not just optimal. We understood more about Arduino and servo and learnt how important microcontrollers are in the control world.

SNo.	Component	Quantity	Price (Rs)	Total
1	Servo Motor MG995	1	500	500
2	Arduino Mega 2560	1	650	650
3	Arduino UNO	1	250	250
4	Ultrasound Sensor SR 04	1	100	100
5	Bluetooth Module HC 05	1	250	250
6	Breadboard	1	70	70
7	Jumper wires	20	2	40
8	3D Printing Filament	70	4.2	294
			Total	2154

Fig. 15 Cost of the components

Fig. 16 The A-Team