Ball and Beam Project
Final Report

EE198 Robotics System Project

Student Name:	Student Number:
Ben Murray	21302753
Donal McCann	21473402
Jordan Browne	21365316
Leah Duggan	21335851
Megan Daly	21318651
Sean Loughins	21344501

Abstract

The ball and beam system is a non-linear, unstable system which is widely used as a benchmark control setup for evaluating various control strategies. The significance of the ball and beam system is that it is a simple system which is open-loop and unstable. If the beam is restricted to be nearly horizontal, without active feedback, it will swing to one side or the other, and the ball will roll off the end of the beam. To stabilize the ball, a PID controller will be which creates a control system which measures the position of the ball and adjusts the beam.

Table of Contents

1. Introduction

The Ball and Beam Balancing System is used in many classrooms to explain control theory or, more specifically, a closed loop system. This project describes the necessary design and construction of a ball and beam balancing system. It also describes the sensor and control system needed to balance the ball on the beam. The purpose of the project is to control the position of the ball using the sensor and control system. To do this the angle of our beam is manipulated, preventing the ball from falling off of the beam. The beam is made of aluminium which provides a track for a ball to freely roll upon. One end of the beam is connected to a motor and the other end is fixed in place. An ultrasonic sensor is used to measure the position of the ball. This information will be sent to the controller causing the motor to react to the position of the ball and attempting to balance the ball upon the beam.

2. Problem at hand

The problem at hand is to balance a ball at a point set by the user without any input from the user. This problem consists of two main parts that must be solved to achieve the successful positioning of the ball. Firstly, the balls position from the setpoint (e.g., the point that the ball is desired to be at) the distance that is found must then be reduced by moving the beam in a way that causes motion in the ball. Secondly the ball must be controlled and sped up/slowed to get it to the setpoint, it must also have its direction controlled as it may need to pass the setpoint multiple times before settling. Once both factors are considered a control system must be designed to measure the balls position, control the movement of both the ball and the beam and reduce the error between the balls position and the setpoint. This must be achieved without any user input to the system as it is expected that the system will operate independently in a closed loop system.

3. Solutions to the problem at hand

3.1 Solution One

The first solution devised was to attach a motor to the beam at exactly the centre of the beam. In other words, at exactly the midpoint of the beam's length. This creates a simple mechanical implementation of the ball and beam. This design allows us to design a simpler controller for the MegaPi, while keeping the complexity of the systems dynamic parts low. The effect of gravity and the slope of the beam are present in this setup however this setup has a chance to not reach the setpoint accurately as the motor may cause overshoot on the ball and there could be a delay in changing the motors direction in response to the ball passing by the motors position. This design also has limited control on the beam as the ball moving position will cause the weight distribution on the beam to change suddenly reducing the control of the motor on the beam as it will need to increase its power to respond accordingly to the change in weight distribution.

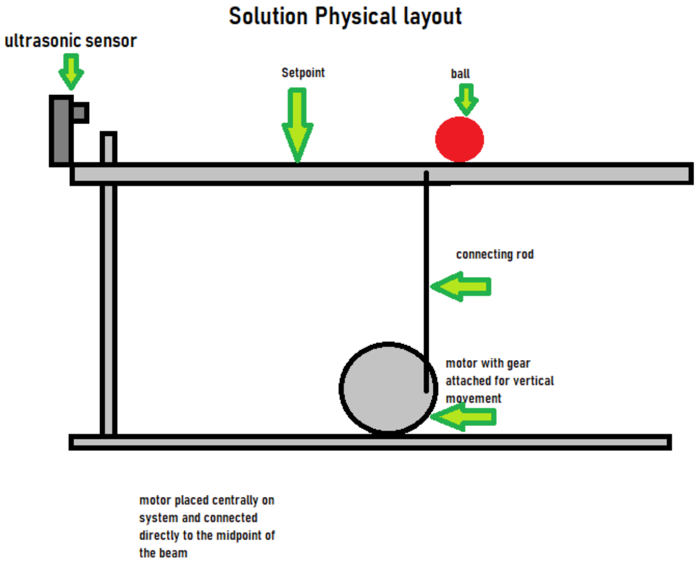


Figure 1.1 First Solution

3.2 Solution Two

The Second solution and the solution implemented is to place the motor, gear and connecting rod directly below the end of the beam and connecting it there. The advantages over the previous system are the greater control over the elevation and depression of the beam as the length of the connecting rod is smaller and the effect of turning the gear is greater when connected directly below the end of the beam. This allows for finer tuned control of the ball along the beam and helps to get the ball to the setpoint more accurately. The disadvantages of this system are the same as the system above, however the problem of ball control from the first system has been solved as well as the problem presented by the movement of the weight of the ball and the position the force of the weight acts on. The complexity of the model of this system setup is like that of the system listed above.

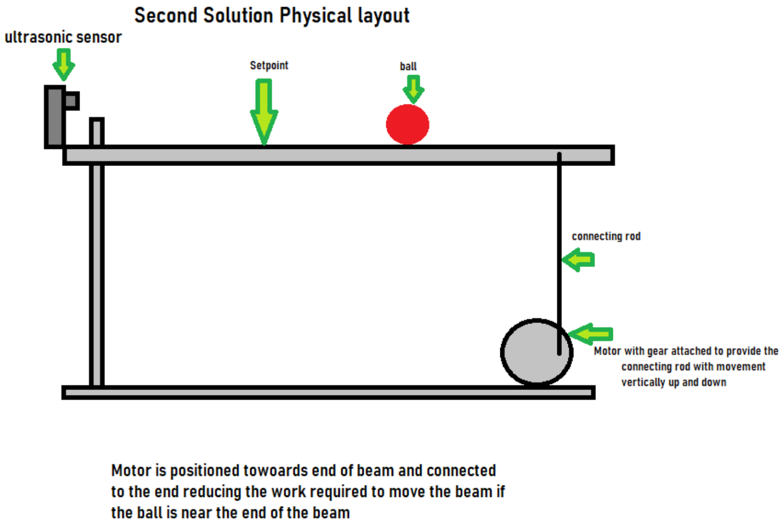


Figure 1.2 Second Solution

4. Research

4.1 The physical Structure

The physical structure of this ball and beam project was designed while keeping in mind that the objective of this project is to build a system using the Makeblock kits. Therefore, the kits would have simple and easy to assemble parts. The first physical aspects of this system that were discussed, and possibly the most important parts, were the frame, sensor mechanisms and the balance beam. The ball and beam project are widely used for educational purposes as it has a non-linear mathematical model which is easy to understand and implement. Figure 2.1 and Figure 2.2 were used as references while trying to understand the model:

Figure 2.1 Model of a Ball and Beam System

Figure 2.2 Model of a Ball and Beam System

The desired set up for this ball and beam system was to use the long aluminium beams included in the Makeblock kits for the ball to roll along. There are two degrees of freedom within this system. The first being the ball rolling along the beam and the second being the beam rotating through its connected axis. The beam would be somewhat V-shaped allowing easy movement of the ball. A lever will be attached to one end of the beam and a gear at the other. The gear turns with an angle of theta θ, while the lever changes the angle of the beam by alpha α. [5] [Figure 2.3]

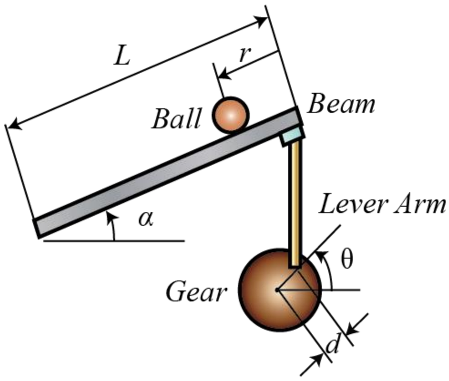


Figure 2.3 Variable Diagram

The sensor used is an ultrasonic sensor. Many different sensors were research for this project. For example, a laser sensor could have been used for this system but ultimately, it was decided that that an ultrasonic sensor would be easier. It is required that the ball remains in contact with the beam, this imposes a restriction on the rotational acceleration of the beam. The system must measure the position of the ball from the ultrasonic sensor and calculate the error signal from the ball to the setpoint. Once this error is found the system applies this error as feedback until the controlled process variable is the same value as the setpoint. This sensor works by generating an ultrasonic wave or pulse and measuring the time it takes for an echo to return.

The motor plays the most important part in this ball and beam system. The motor changes electrical energy into mechanical energy. The sensor sends the position of the ball to the Arduino and the motor will turn to match the position necessary to keep the ball balanced on the beam.

This system deals with a closed loop system. A closed loop system is also known as a feedback control system. A feedback control system is one that tends to maintain a prescribed relationship between one system variable and another. It compares functions of these variables and uses the difference as a means of control. In its basic format, the feedback control system has three important elements: the actuator, the sensor, and the controller. [Figure 2.4]

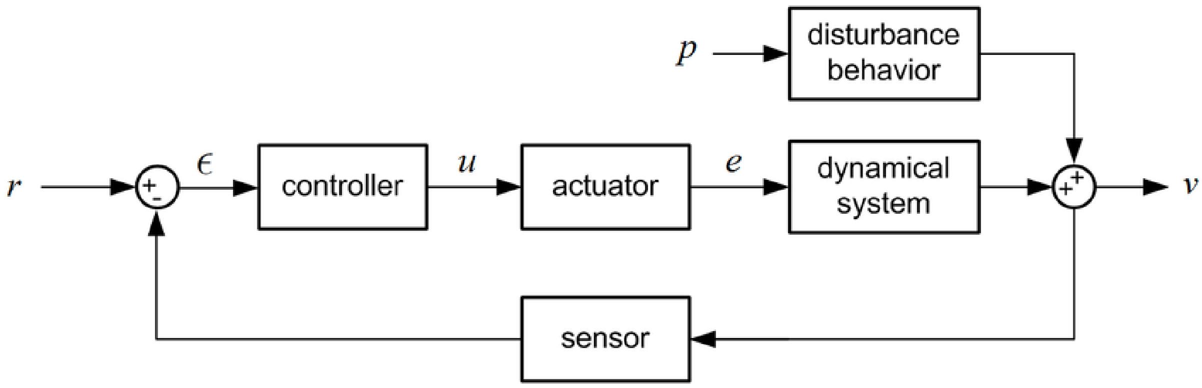


Figure 2.4 Block Diagram of a Control System

The actuator is the mechanism that changes the input to the plant. The sensor is the device used to measure the output. An ultrasonic sensor is used in this system to measure the position of the ball from the desired setpoint. The controller is the device that outputs an appropriate control signal based on the difference between the measured output and the desired setpoint. In relation to this system, a PID controller is used to obtain appropriate responses to the position of the ball. PID stands for Proportional-Integral-Derivative. Each of these elements are multiplied by its value and the sum is used to determine the new inputs for the actuator.[8]

4.2 The mathematical structure

The mathematic model for the ball and beam was derived using Newtonian physical laws
This formula contains three forces, the first term deconstructs the formula for the inertia of a cylinder, a cylinder was use because the ball's axis of rotation is restricted by the beam, the second term is describing the force due to gravity and this is multiplied by a coefficient between 0 and 1 depending on the angle of the beam and then the third force is the centripetal force which is proportional to the angular velocity squared and is in the opposite direction of the traditional force. In this 'r' represent the position of the ball on the beam. Simulink and MATLAB were used to create a model that can be manipulated using block diagrams and then the ball's reaction was plotted against time.

4.3 Measurement of the ball's location on the beam

The location of the ball must be measured accurately as to avoid any errors when it arrives at the setpoint. The sensors that can be used are ultrasonic sensors, laser rangefinders and various other measuring devices. The laser rangefinder uses a laser beam and its reflection of light from the ball to determine the distance from it to the ball. The sensor used on our system is an ultrasonic sensor as it provides an accurate measurement for the distance from the sensor to the ball as well as being compact and low power consumption when compared to a laser rangefinder. The ultrasonic sensor uses ultrasonic waves to find the position of the ball by measuring the time it takes for the ultrasonic wave to return to the sensor once it reflects off the ball. By dividing the time taken for the wave to return to the sensor by two and using the speed of sound in air, the distance can then be calculated.
This distance can be calculated using the formula: [Equation 1]

$$d = \frac{1}{2}TC$$

Equation 1.1

Where T is the time for the wave to return to the sensor and C is the speed of the ultrasonic wave. To find the distance between the ball and sensor the formula must be multiplied by ½ as the result without this would be the distance from the sensor to the ball and back to the sensor as the sensor must use the waves reflection from the ball to measure the distance. E.g., the waves echo off the ball.

5. Modelling of the System in MATLAB and Simulink

5.1 Mathematical Derivation

$$mr'' = mgsin(\alpha) \text{ [Equation 2]}$$

Equation 2 is the force equation for the ball, the force on the ball due to gravity is multiplied by the coefficient sin(α) which is 0 when the beam is flat indicating no force in the direction of the beam, and 1 when the beam is vertical. mass can be cancelled out both sides and left with a simple equation 2.1

$$r'' = gsin(\alpha) \text{ [Equation 2.1].}$$

As α is the beam angle it needs to be written in terms of the motor angle θ. the formula relating θ to α is

$$\theta = \sin^{-1}\left(\frac{d}{l}\sin(\alpha)\right) \text{ [Equation 3],}$$

substituting equation 3 into equation 2 gives

$$mr'' = \frac{d}{l}gsin(\theta) \text{ [Equation 2.2]}$$

as sine is not a linear function a linear approximation of sin(θ) is sin(θ) = θ around θ = 0, a Taylor series was considered for approximating sin(θ) but the lowest order transfer function it would give is a third order function as the second term of the Taylor series is to the third power which can be seen in equation 4.

$$\text{Sin } (\theta) \approx (\theta - \frac{\theta^3}{3}) \text{ [Equation 4]}$$

Now the transfer function of Equation 2.2

$$r'' = \frac{d}{l} g\theta \text{ [Equation 2.2]}$$

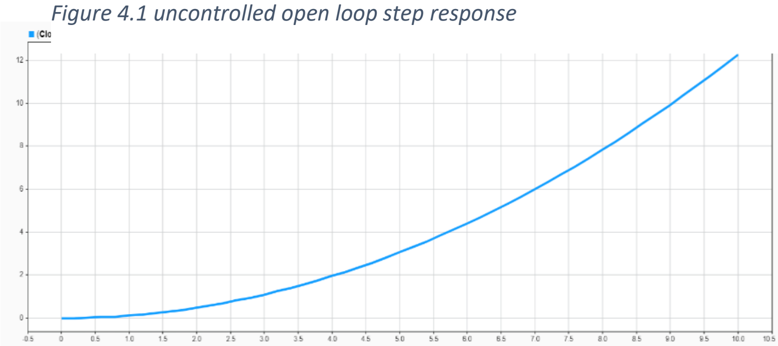
can be found by taking the Laplace transform of the input and output and taking no initial conditions gives the transfer function

$$\frac{R(s)}{\theta(s)} = \frac{1}{L}\frac{dg}{s\text{^}2} \text{ [Equation 5].}$$

5.2 Analysis

Filling in the given values and solving for the poles, the poles are found at the origin and because the poles are not exclusively left of the imaginary axis this means that the system is not inherently stable which can be seen in figure 4.1.

Figure 4.1 uncontrolled open loop step response



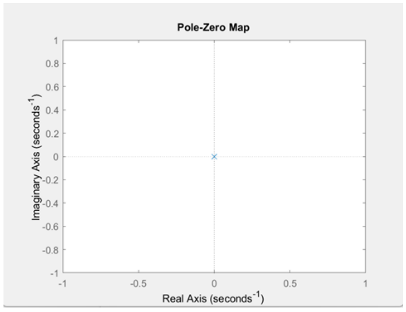


Figure 4.2 Pole zero map

In figure 4.2 we can see the step response of the transfer function [Equation 5] the graph shows how the system becomes unstable and the ball will roll off the end of the beam in approximately three seconds.

5.3 Controller Design

In figure 4.3 the block diagram of the controlled system can be seen. the output of the transfer function is the current position of the ball and the input of the transfer function is the angle θ, the input to the transfer function is controlled by a P.I.D controller, The P.I.D controller uses the summation of three signals which then is fed into the transfer function [Equation 5] the first

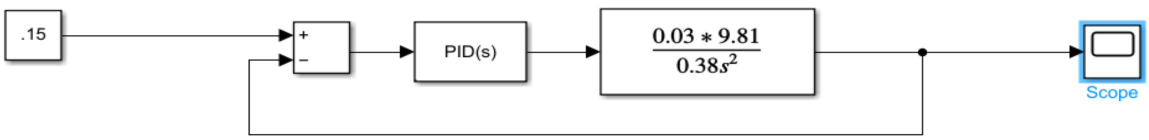


Figure 4.3 block diagram of controlled system

signal (P) is a signal that is just proportional to the error signal, this will help correct a larger error using a greater change in θ and this is multiplied by a proportional gain K_p and the second signal (I) is proportional to the integral of the error over time and is multiplied by its gain K_I

this signal eliminates steady state error in the system because if the sum of error is increasing this means that there is a steady state error and the reaction of the system will grow larger as the steady state error exist and finally the third signal (D) is proportional to the change of error which will help reduce overshoot in the system and acts as a damper in the system[6], the P.I.D controller modelled as a block diagram can be seen in figure 4.4.

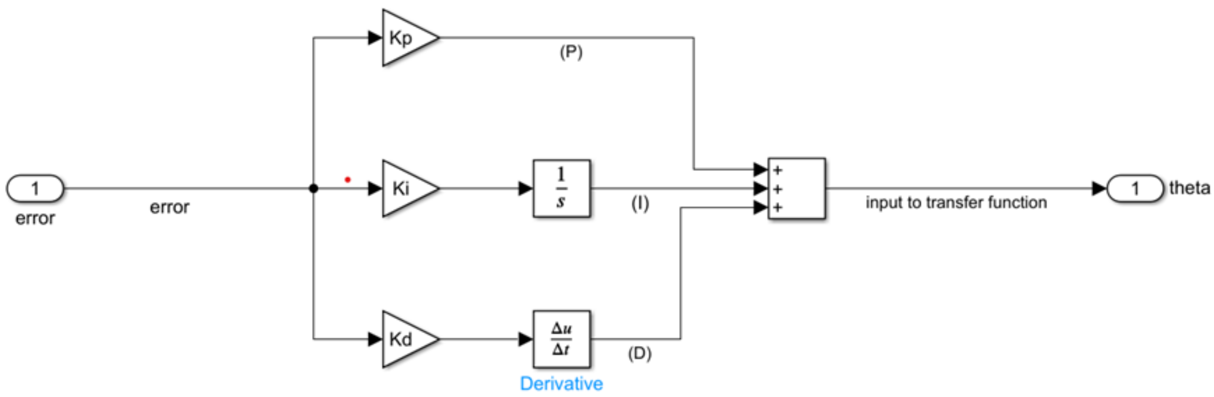


Figure 4.4 Block Diagram of PID Controller

5.4 Choosing Values for KP, KI and KD

There are many ways to choose the gain values in a P.I.D controller; [Table 1] the most common way is known as the Ziegler-Nichols method. Step one is to set the Integral gain and derivative gain to zero and raise the proportional gain until a sinusoidal wave can be seen [Figure 5.1], this value of Proportional gain is called K_c and for this system a K_c of 10 was used, by analysing the wave output on the scope the T_c value to be used in the calculations is 2.26 seconds Then using [1] the optimal values for K_p , K_I, and K_D can be calculated and it was found to be: K_p = 6, K_I = 5.3, K_D = 0.17. The controlled system using these values can be seen on Figure 4.6. The Ziegler-Nichols tuning method does not give an adequate response and the system remains unstable, by analysing Figure 4.6 it is observed that proportional gain seems to be too low as the ball is not reponding to the proportional gain and becoming eccentric, and the derivative gain is also too low

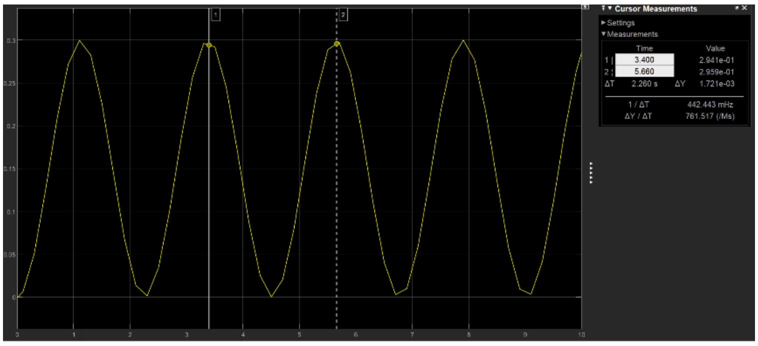


Figure 4.5 sinusoidal wave given by Kp = 10

Control Type	K_P	K_I	K_D
P	$0.50K_c$		
PI	$0.45K_c$	$1.2K_p/T_c$	
PID	$0.60K_c$	$2K_p/T_c$	$K_pT_c/8$

Table 1 Ziegler-Nichols tuning table

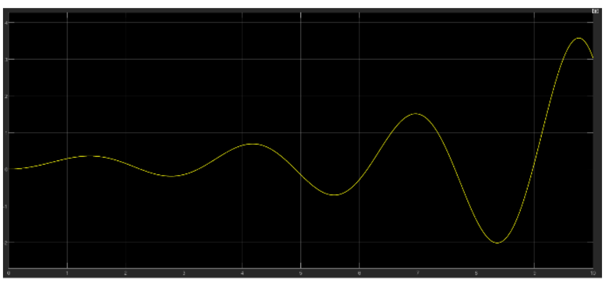


Figure 4.6 System Response from Ziegler-Nichols Tuning

for this system, because of the overshoot that Is still happening and also becoming more extreme. Manually altering these values seems to be the next option going forward, adding these new manual gains and observing the step response can be observed in Figure 4.7, the new gains were set to Kp = 15, Ki = 5.3, Kd = 10.

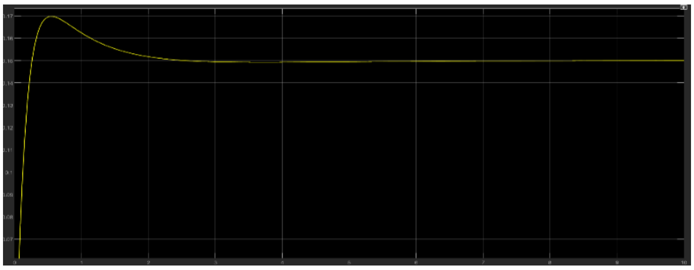


Figure 4.7 System response from manual tuning

5.5 analysing the controller

A good start to analyse the system is to find the closed loop transfer function, to do this, we need to find the transfer function representation of a PID controller which can be seen in Equation 6, the formula for finding a closed loop transfer function is $CLTF = \frac{G}{1+GH}$ [Equation 7] which relates to the block diagram found in figure 4.8, using this the closed loop transfer function was calculated to be: $\frac{(Kd)(d)(g)s^2+(Kp)(d)(g)s+(Ki)(d)(g)}{Ls^3+(Kd)(d)(g)s^2+(Kp)(d)(g)s+(d)(g)}$[Equation 8].

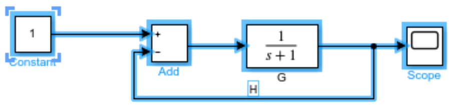


Figure 4.8

$$\frac{K_d s^2 + K_p s + K_i}{s}$$

Equation 6 transfer function of PID controller

Filling in the values of d and L based on the system characteristics from table 2, the CLTF of the entire controlled system is $\frac{2.943s^2+2.94s+1.94}{s^3+2.94s^2+2.94s+1.96}$[Equation 9] the step response of this block is seen in figure 4.8.

d	0.02
L	0.38
g	9.81

Table [2] System characteristics

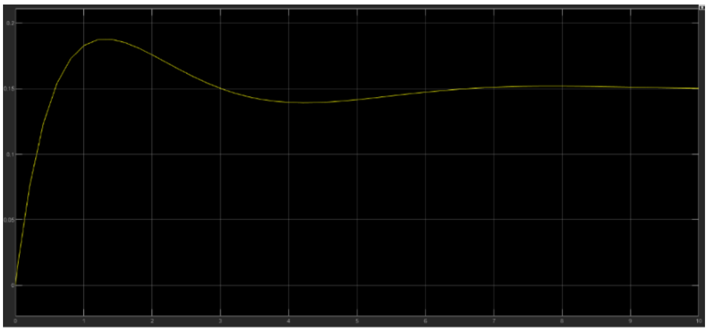


Figure 4.9 CLTF step response

Analysing the closed loop transfer function, it is observed that the settling time is 6 seconds and the percentage overshoot is 40%.

5.6 Where does this Model Fail?

When the linear approximation of sin(θ) was made to be sin(θ) = θ this model can break down around the extremes because this approximation is centred around θ = 0, which can be seen in figure 6.5.

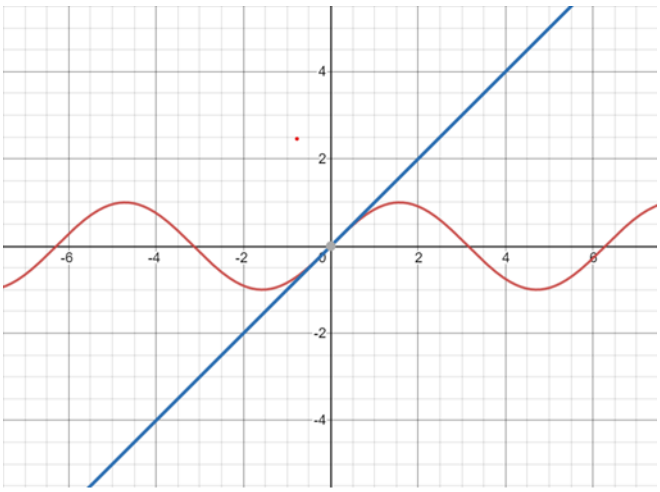


Figure 6.5 plotting of ϑ

6. System Characteristics

The final setup for the system used a 38cm beam for the ball to roll along. The ultrasonic sensor was placed at one end of this beam. An issue occurred where the ball would hit the ultrasonic sensor causing the system to go out of control and continuously spin around. To fix this issue a stopper was placed 5cm in front of the sensor. This also helped to prevent any damages occurring to the ultrasonic sensor. Below the sensor there is a motor and a gear to allow the beam to move. The length of the lever doesn’t affect this system as it will only move by the 4cm diameter of the gear. At the other end of the beam, we placed a weight. This was to help balance out the beam as there were issues with the motor coming under increased strain due to the weight of the ball acting on the furthest point of the beam caused the motor to need to increase its power output causing the system to overshoot the setpoint multiple times.

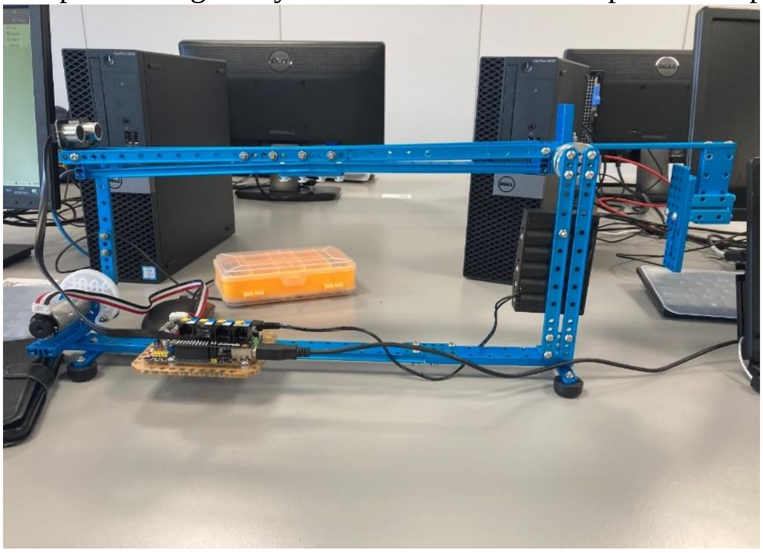


Figure 7 The Ball and Beam System

7. Programming the system

The system programming was done in the Arduino IDE, the MegaPi is equivalent to the Arduino Mega 2560 therefore that Board should be selected in the settings of the IDE, to calculate the derivative of the error, calculate two error signals with a delay of 50 milliseconds in between them, a delay of 50 milliseconds was chosen because this offers the least system response lag, and still a calculatable derivative because the most accurate the ultrasonic sensor can measure is to 0.001m, therefore a reasonable speed of the ball wont incur a change between the two measurements, to calculate the integral add the error onto the integral variable for each loop of the code and multiply by the time taken between each measurement which in this case is 50 milliseconds due to the delay in the system from the derivative. When calculating the error, the ultrasonic sensor does not consider the fact the balls centre of mass is in the centre of the ball approximately two centimetres further than what the ultrasonic sensor measured, this is described in figure 8.1

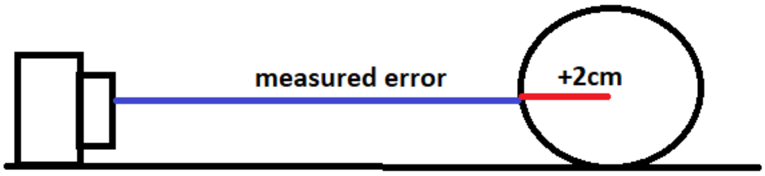


Figure 8.1 error fixing

Makeblock provides a library which makes coding it a lot simpler, as all the necessary objects are already created, which include getting the motor to rotate by a certain amount, the encoded motor does not store its absolute position, only its relative position in relation to when it was turned on, a problem that occurred due to this was that the motor would turn by its desired angle not “to” its desired angle, the solution for this is subtracting the desired position and the current position this method rotates the motor to a given position for the given error values, the code for this can be seen in figure 8.2

```
Encoder_1.move((((DerivationGain * derivative + proportionalGain * error2 + IntegralGain
* integral)) - Encoder_1.getCurPos()),abs(300));
```

Figure 8.2, code to move encoded motor to a desired angle

The gains chosen for the model of the system were too high and results in the motor spinning too far which is not ideal, therefore its best to choose new gains, but the ratio of the gains to each other is still valid. Its best to start off finding a derivative gain that works, to do this set Kp and Ki to 0 and change Kd until the system kills the motion of the ball, when that happens then choose the proportional gain based off the ratio of the Kd to Kp that was found in the model. The chosen Kd value was 2.2 and then based off of the model Kp should be around 1.32 and the Ki = 0.77, in testing it was found that a lower Kp of 0.7 kept the system more stable because if the ball got too fast which would happen with a steeper incline the derivative gain pushes the gear past ± 90 degrees making the movement null as this would make the beam come back down/up.

8. Ethical considerations

Energy is vital part in considering the ethical side of engineering as many [1], if not most of the energy in the world is unsustainable and damaging to the planet when we were designing our ball and beam project, the whole system is powered by six alkaline batteries, these batteries are generally safe for the environment due to their protective casing, over time the batteries in the landfills can contaminate ground water once their casing wears down, however, this is only a concern if the batteries are inappropriately disposed of. Health and safety while building with the Make-Block was also considered [2], even though it was relatively safe, there were pinch point identified on the moving parts of the system, i.e., at the hinge of the beam and near the gear on the servo motor, the maximum torque of the servo is five kilograms this means that the system is relatively safe.

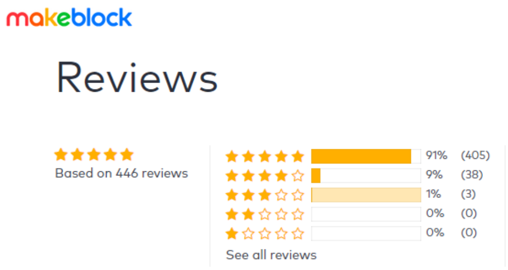


Figure 7.1 Makeblock reviews

9. Future work

The make-block kit contains a microcontroller that was used for programming the system, this controller's brand is MegaPi, the model is a MEGA 2560, it is compatible with the Arduino coding environment. The Arduino programming language is a new skill each member of the team has become familiar with and this can be used in future projects. In regard to what else can be done with this particular system, the system can be tested and further improved, the team will be allocated responsibility according to their strongest skills and how they apply to the project. Basic improvements to the response time of the system by calculating a higher accuracy gain value will also be carried out. If this system was to be redone, it is possible to reduce the part count and make the system less bulky and easier to assemble.

10.Conclusions

For this project a system to Control a balls position in relation to a user inputted setpoint was required to be designed both in a Simulink model for analysis as well as a physical real-world model for testing the hardware and software. Firstly, the group decided on two solutions to investigate. While both had advantages and disadvantages the group decided to implement Solution two which is detailed above in the problem at hand section. Once this solution was chosen the group was broke into two, one team assembled the physical model of the system from solution two while the other team worked on modelling the system in Simulink.

The Simulink model was needed to help with analysing the systems response to move the ball to the setpoint, as without an accurate simulation the group would not have had a mathematical model accurate enough for implementation into the hardware. Upon completion of the Simulink model the response time, overshoot percentage, two percent settling time and the stability of the system. These values are response time = 3sec, overshoot percentage = 20%, two percent settling time = 2.1 sec and a marginally stable system is discovered as the graph has the characteristics of a marginally stable system E.G it has limited oscillation along the y axis, the system settles quickly and exactly at the setpoint.

The physical model was assembled from parts available in the make block kits provided to the group. From this kit the following components were utilised to make the system 1x MegaPi 2560 board, 1x shield, 1x ultrasonic sensor, 1x encoder motor. Once assembled by the other team the group reformed while utilising the simulation to help in the programming of the systems code the group was able to achieve the desired goal of the ball being moved to the setpoint and settling on that point accurately. This was first achieved with a slow movement of the system to test if the code worked as anticipated, once the system was found to work the group then proceeded to adjust and fine tune the system to increase its speed and accuracy. While the system was only needed to respond accurately improving the system would allow the group to work on the skills required to work in some if not all industry situations. However, there was an issue present in the beam of the system as a small gap was present between two of the sections of the beam that can cause the ball to stop abruptly in this gap if the setpoint is close to this gap. Upon testing the system, it was discovered that the radius of the ball must be included in the calculations as it causes an inaccuracy in the calculations as the sensor detects the closest point to it which would be approximately the length of the radius from the centre of the ball.

11. References

[1]2022. [online] Available at: https://globalrecycle.net/are-alkaline-batteries-hazardous-waste/ [Accessed 10 March 2022].

[2]Makeblock. 2022. MegaPi - Makeblock. [online] Available at: https://www.makeblock.com/project/megapi#:~:text=MegaPi%20is%20a%20main%20control%20board%20specially%20designed,2560%20and%20supports%20programming%20with%20Arduino%20IDE%20perfectly [Accessed 10 March 2022].

[3]Makeblock. 2022. Reviews. [online] Available at: https://store.makeblock.com/pages/reviews [Accessed 10 March 2022].

[4]Nbleisonmotor.com. 2022. [online] Available at: https://www.nbleisonmotor.com/LS-25GA370-Dc-Gear-Motor-pd6387204.html [Accessed 10 March 2022].

[5]2022. [online] Available at: https://www.researchgate.net/publication/284187009_Ball_and_Beam_Control_System [Accessed 4 May 2022].

[6]Instrumentation and Control Engineering. 2022. *What is proportional, integral and derivative control? | Instrumentation and Control Engineering*. [online] Available at: https://automationforum.co/proportional-integral-derivative-control / [Accessed 4 May 2022].

[7]Mechatronicstutorials.blogspot.com. 2022. *Balancing of a Ball on Beam using Arduino as a PID controller*. [online] Available at:

http://mechatronicstutorials.blogspot.com/2014/07/balancing-of-ball-on-beam-using-arduino.html [Accessed 4 May 2022].

[8]Swarthmore.edu. 2022. [online] Available at: https://www.swarthmore.edu/sites/default/files/assets/documents/engineering/ww_final_e90.pdf [Accessed 4 May 2022].

12. Risk Assessment Form

Department of Electronic Engineering, Maynooth University

PBL PROJECT RISK ASSESSMENT FORM

NAME AND STUDENT NUMBER: Group 4: Ben Murray (21302753), Donal McCann (21473460), Jordan Browne (21365316), Leah Duggan (21335851), Megan Daly (21318651), Sean Loughins (21344501)	PROJECT NAME: Ball and Beam System
SUPERVISOR: John Dooley	PROJECT LOCATION: Various labs, see the PBL schedule for details.

BRIEF DESCRIPTION OF PROJECT: to control the position of the ball along a beam using the sensor and control system.

Hazards, Risk [High(H) Medium (M) Low (L)], and Control Measures

HAZARD	Risk	Controls
Electrical work	L	system runs off a 5V battery pack
Mechanical work	L	Light mechanical work may be required, ie bolting of metal segments, some bending of metal to shape. If needed access to the workshop will be done under the workshop guidelines.

Identified risks should be discussed with your supervisor and a safe system of work agreed. A more in depth risk assessment may be required after initial review. **Do not proceed until form is signed off.**

Further Controls Required

None

Note: that this is not currently and exhaustive list of potential risks. As risks are identified they will be added to this document and assessed.

SIGNATURE OF STUDENT: Ben Murray DATE: 05/05/22

SIGNATURE OF STUDENT: Leah Duggan DATE: 05/05/22

SIGNATURE OF STUDENT: Sean Loughins DATE: 05/05/22

SIGNATURE OF STUDENT: Donal McCann DATE: 05/05/22

SIGNATURE OF STUDENT: Jordan Browne DATE: 05/05/22

SIGNATURE OF SUPERVISOR: ____________ DATE: ________

DEPT HEALTH AND SAFTY OFFICER: ____________ DATE: ________

13.Appendix

Arduino code

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include <MeMegaPi.h>

MeUltrasonicSensor ultrasonic_7(7);
double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
MeEncoderOnBoard Encoder_1(SLOT1);
MeEncoderOnBoard Encoder_2(SLOT2);
MeEncoderOnBoard Encoder_3(SLOT3);
MeEncoderOnBoard Encoder_4(SLOT4);

float proportionalGain = 0; //initialising all variables to be used
float setpoint = 0;
float DerivationGain = 0;
float error1 = 0;
float error2 = 0;
float derivative = 0;
float IntegralGain = 0;
float integral = 0;

void isr_process_encoder1(void){
  if(digitalRead(Encoder_1.getPortB()) == 0){
    Encoder_1.pulsePosMinus();
  }else{
    Encoder_1.pulsePosPlus();
  }
```

```
}

void isr_process_encoder2(void){
  if(digitalRead(Encoder_2.getPortB()) == 0){
    Encoder_2.pulsePosMinus();
  }else{
    Encoder_2.pulsePosPlus();
  }
}

void isr_process_encoder3(void){
  if(digitalRead(Encoder_3.getPortB()) == 0){
    Encoder_3.pulsePosMinus();
  }else{
    Encoder_3.pulsePosPlus();
  }
}

void isr_process_encoder4(void){
  if(digitalRead(Encoder_4.getPortB()) == 0){
    Encoder_4.pulsePosMinus();
  }else{
    Encoder_4.pulsePosPlus();
  }
}

void _delay(float seconds) {
  if(seconds < 0.0){
    seconds = 0.0;
  }
```

```
  long endTime = millis() + seconds * 1000;
  while(millis() < endTime) _loop();
}

void setup() {
  Serial.begin(9600);
  attachInterrupt(Encoder_1.getIntNum(), isr_process_encoder1, RISING);
  Encoder_1.setPulse(8);
  Encoder_1.setRatio(46.67);
  Encoder_1.setPosPid(1.8,0,1.2);
  Encoder_1.setSpeedPid(0.18,0,0);
  TCCR1A = _BV(WGM10);
  TCCR1B = _BV(CS11) | _BV(WGM12);
  TCCR2A = _BV(WGM21) | _BV(WGM20);
  TCCR2B = _BV(CS21);
  proportionalGain = 0.2;
  setpoint = 25;
  DerivationGain = 2.2;
  IntegralGain = 0;
  integral = 0;

}

void _loop() {
  Encoder_1.loop();
}

void loop() {
  error1 = ((setpoint - ultrasonic_7.distanceCm())+0.02);
  delay(50);
```

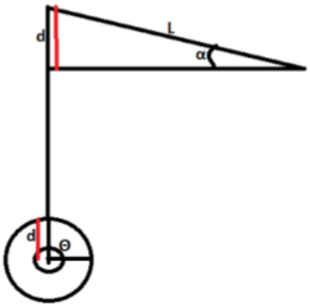
```
  error2 = ((setpoint - ultrasonic_7.distanceCm())+0.02);
  derivative = (((error2 - error1)) / 0.05);


  Encoder_1.move((((DerivationGain * derivative + proportionalGain * error2 + IntegralGain
* integral)) - Encoder_1.getCurPos()),abs(300));
  integral += error2 * 0.1;
  Serial.print("integral ");
  Serial.println(integral);
  Serial.print("derivative ");
  Serial.println(derivative);
  Serial.print("error1: ");
  Serial.println(error1);
  Serial.print("error2: ");
  Serial.println(error2);
  _loop();
}
```

All mathematical derivation

Derivation of α relating to θ:

$$\sin(\alpha) = \frac{h}{L} \quad (h\ is\ height\ of\ the\ end\ of\ the\ beam, which\ is\ equal\ to\ d\ when\ \theta\ is\ 90\ degree\ )$$



Therefore: h = dsin(θ)

Therefore: sin(α) = $\frac{dsin(\theta)}{L}$

Therefore: α = sin^{-1}($\frac{dsin(\theta)}{L}$)

Derivation of closed loop transfer function:

$$\text{Negative feedback loop formula} = \frac{G}{1+GH}$$

$$\text{System TF} = \frac{dg}{Ls^2}$$

$$\text{PID TF} = (Kd)s + (Kp) + \frac{(Ki)}{s}$$

$$\text{System TF * PID TF = G} = \frac{(Kd)(d)(g)s+(kp)(d)(g)+\frac{(Ki)(d)(g)}{s}}{Ls^2}$$

$$\text{Using negative feedback loop formula: } = \frac{(kd)(d)(g)s+(kp)(d)(g)+\frac{(Ki)(d)(g)}{s}}{Ls^2+(kd)(d)(g)s+(kp)(d)(g)+\frac{(Ki)(d)(g)}{s}} \bullet \frac{s}{s}$$

$$\text{CLTF} = \frac{(Kd)(d)(g)s^2+(Kp)(d)(g)s+(Ki)(d)(g)}{Ls^3+(Kd)(d)(g)s^2+(Kp)(d)(g)s+(Ki)(d)(g)}$$