Ball and Beam System Final Report

Module Code: EE198

Authors: Sarah White, Nathan Ewnetu, Tadhg Stones, Eoin Keatley & Edvinas Ivanovas

Abstract:

The ball-and-beam system is a control system that effectively balances a ball on a beam by controlling the angle of the beam with a motor input. For the system to stabilise the ball a feedback control loop was designed. This project included creating a valid system model, identifying model parameters, designing an effective control system, assembling the system hardware, and implementing the control system using an Arduino to interface with the motor. The difficulties of operating the system were discovered using the MATLAB animation program. The ball-and-beam issue is a common nonlinear control application that many people use to show the efficiency of different control approaches.

Acknowledgements:

We would like to thank everyone that took part in the completion of this project, especially to Dr. John Dooley who has provided us with all the necessary material required for the completion of the project.

Declaration:

We hereby confirm that the work presented in this final report is all our own work and any sources that were used are correctly acknowledged and were cited appropriately.

Signed: *Sarah White, Tadhg Stones, Nathan Ewnetu, Eoin Keatley, Edvinas Ivanovas*

Table of Contents

Introductions:

Control engineering is essential to the enhancement of technology. The benefits of an improving control industry can be immense. It can lower production costs, energy consumption and reduce pollution [1]. The Ball and Beam system is one that gives exposure to the basics of a non-linear control system and therefore provides a more in-depth perception of these controllers. This system's objective is to establish a control loop that allows a ball to balance at a certain position on the beam. The procedure is simple to follow. The motor is coupled to the beam in such a way that the motor's shaft may adjust the beam's angle. This is commonly accomplished by attaching a lever arm to the beam's end or simply connecting the motor shaft to the beam's centre [2]. A PID controller was used in the system for this project to maximise efficiency. This type of controller is used in control loop feedback mechanisms. It works by reading the error of the system and calculating an appropriate actuator command. PID stands for **P**roportional **I**ntegral **D**erivative, these three parts of the controller work together to get a steady state error which is as close to the desired output as possible. The error is the difference between the desired position of the ball and the position obtained. The Proportional controller responds to the error at the time it occurs, the Integral controller recalls prior errors, and the Derivative controller forecasts future errors. Flow rate, speed, pressure, and other physical characteristics can all be measured with PID controllers. There are many various types of controllers, but the PID controller was chosen because it is more analytical and considers the future, past, and current information. The PID controller was constructed and tested in MATLAB and Simulink, and the values for P, I, and D were calculated based on the findings.

Section 1: Comparison of Different Designs:

In this section two different designs have been shown that will prove to be the best candidates for the design in this project. The reasons are broken down into four different sections.

1. Setup of Each System
2. Complexity of Design Elements
3. Final Design Chosen
4. Hardware Required for Chosen Design

Section 1.1: Set up of the Wheel and Lever Design

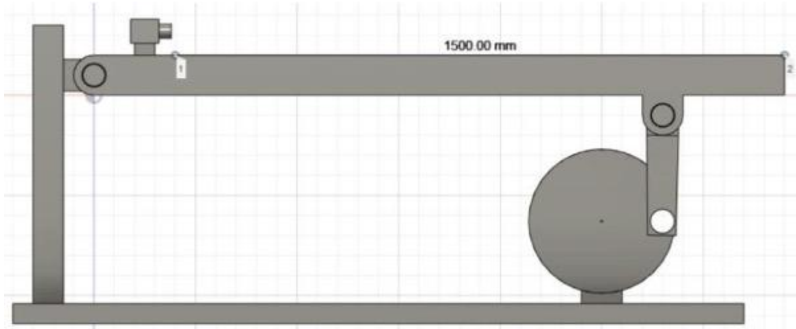


Figure 1

The design in *Figure 1*, was the first design that was encountered during the research phase. The “Wheel and Lever” design uses the end of the beam as a pivot point for adjusting the angle. The pivot point is attached to a large gear by a lever. The large gear is connected to the motor which allows it to rotate and counteract the ball's movements on the beam. Due to the distance from the centre of the gear pin, the motor will have to exert more force to turn.

Section 1.2: Set up of the Centre Balance Design

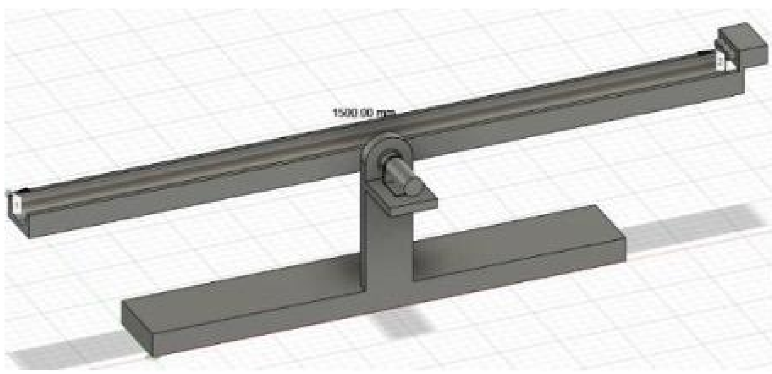


Figure 2

In this design, *Figure 2*, the pivot point of the beam is directly in the centre, this was the rarest of the designs found in our research as many considered the “Wheel and Lever” design more common and as a result be able to be replicated in a more accurate way. At face value the design appears to be less complex than the “Wheel and Lever” due to the exclusion of a gear and pole rotational system. This system opts for a more direct approach of having the motor centred and will require less rotational torque to turn due to the motor only having to lift one side at any given time to adjust the balls position on the beam.

Section 1.3: Complexity of the Wheel and Pole Design

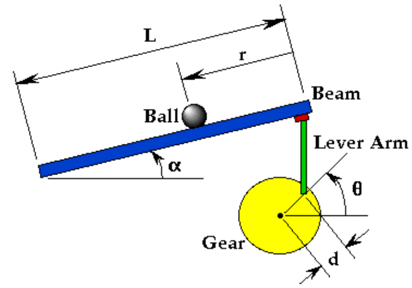


Figure 3

This design had many variables to account for like L - Length of the beam. r - the balls position on the beam, R - the radius of the ball, d - the distance between the centre of the gear and the join between the lever arms and θ - the beam angle. Other variables to be accounted for would be the rotational angle of the gear. While researching it was found that the Lagrangian equation would be needed for this design. This Langrangian equation used was: $0 = (\frac{J}{R^2} + m)r + mgsin\alpha - mr(\alpha)^2$. This equation helps find the motion of the ball on the beam and derive more complex equations such as the Laplace Transform and the Transfer Function equations for the system.

Section 1.4: Complexity of the Centre Balance Design

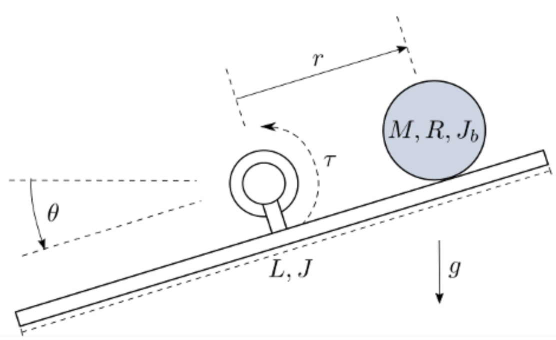


Figure 4

This design had many variables to be accounted for such as M - Mass of the ball, R - Radius of the ball, J_b - Moment of inertia of the ball, r - Balls position on the beam, L - Length of the beam, J - Moment of inertia and θ - Deflection from the horizontal line [3]. This design would mean more power needed by the motor to be able to move the beam on by itself. Both designs were similar in physical technical design which made the choice slightly harder.

Section 1.5: Final Design

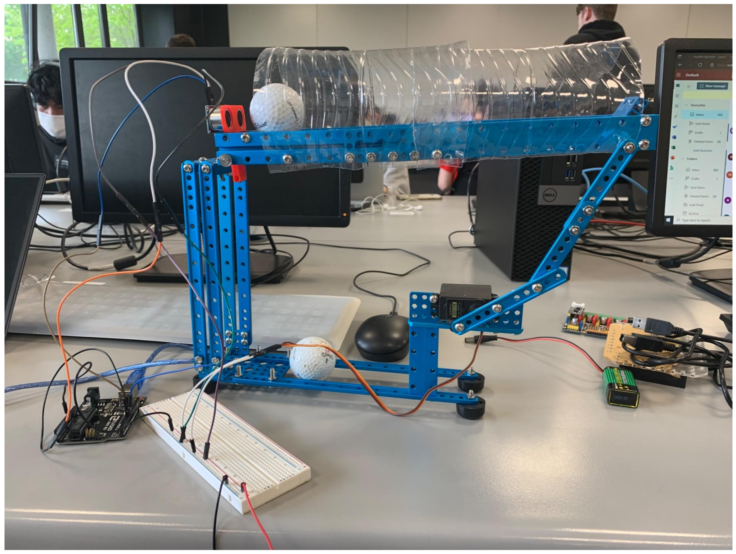


Figure 5

The final design *(Figure 5)* chosen by the group, after careful consideration, was an adaptation of the "Wheel and Lever" design. This design was chosen due to the vast amount of information provided during the research phase of the project. The group felt if there were any difficulties throughout the project, the information researched would be sufficient in helping find a solution to any problems. The design adjusts the angle of the beam by using the end of the beam as a pivot point. A lever connects the pivot point to another lever. The motor is connected to this lever, which allows it to rotate and counterbalance the motions of the ball on the beam. An empty bottle was attached to the beam in order to prevent interference affecting the sensor i.e., external sound waves.

Section 1.6: Hardware Used in the Final Design

One of the key tiebreakers for the decision of the design was the choice of hardware and the placement of such. There are three major requirements for hardware implementation.

1. Ultrasonic Sensor
2. Servo motor
3. Arduino

For the final design, the hardware that was chosen to be implemented was based on its simplicity and its relative familiarity with the group.

Ultrasonic Sensor:

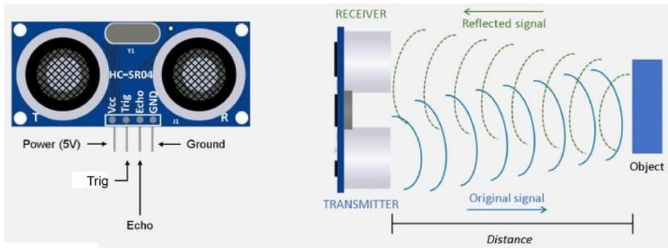


Figure 6

This is an Ultrasonic Sensor *(Figure 6).* This sensor uses ultrasonic waves to determine a distance and return a value based on that distance. It is relatively simple to set up and is fully compatible with Arduino and Mego-Pi.

Servo Motor:

Figure 7

This is a Servo Motor *(Figure 7)*. It is a self-contained electrical device that rotates the lever, connected to the beam, with high precision and efficiency. This motor's output shaft can be changed to a specific angle, location, and velocity that a standard motor cannot. An ordinary motor is used in the servo motor, which is coupled with a sensor for positional feedback.

Arduino:

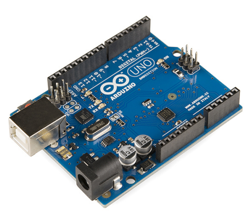


Figure 8

This is an Arduino *(Figure 8)*. An Arduino consists of both a physical programmable circuit board, often referred to as a microcontroller, and a piece of software, or IDE (Integrated Development Environment) that runs on a computer which is used to write and upload computer code to the physical board.

Section 2: Theory:

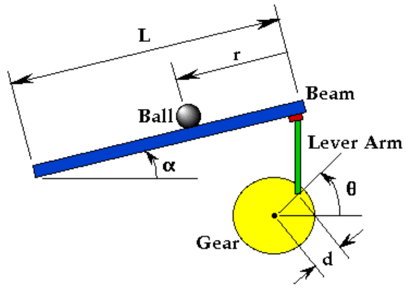


Figure 9

There was extensive research completed on the mathematics and physics that is required to meet the brief given. The mathematics varied depending on the design that was decided. The design is the “The Wheel and Lever” Design *(Figure 9)* as on a mathematics, physics, and construction basis there is a more straightforward understanding required. To begin understanding the mathematics for the design it is vital to write down and understand necessary equations for future mathematical problems.

Section 2.1: Mathematics of System

The following mathematics is required to help find the solution to the given problem, which is to balance a ball on a beam without it falling off [4].

Section 2.2: Variables and Constants

1. m = Mass of the ball → 0.24kg
2. R = Radius of the ball → 0.04m
3. d = Lever arm offset → 0.045m
4. g = Gravitational acceleration → $9.8m/s^2$
5. L = Length of the beam → 1.5m
6. J = Balls moment of inertia → $8.64e^{-3}kgm^2$
7. r = Displacement for the ball → 0.3m
8. α = Beam angle of elevation
9. θ = Servo gear angle

Note:

- Setting time = < 3 seconds
- Overshoot = < 5%

Section 2.3: System Equations

Lagrangian equation of motion for the ball [5] for the chosen system, seen in *Figure 8*:

$$0 = (\frac{J}{R^2} + m)r + mgsin\alpha - mr(\alpha)^2 \qquad (1)$$

The following linear approximation of the system is obtained by linearising the equation about the beam angle, α = 0:

$$(\frac{J}{R^2} + m)r = - mg\alpha \qquad (2)$$

Linearizing equations is the process of modifying an equation to produce new variables which can be plotted to produce a straight line graph.

The equation that connects the beam angle to the gear angle can be approximated as linear using the following formula:

$$\alpha = \frac{d}{L}\theta \qquad (3)$$

By substituting equation (3) into equation (2), we get:

$$(\frac{J}{R^2} + m)r = - mg\frac{d}{L}\theta \qquad (4)$$

Section 2.4: Transfer Function

In this section we use equations (1) to (4) to calculate the Laplace Transform and the Transform Function.
Laplace Transform of equation (4):

$$(\frac{J}{R^2} + m)R(s)s^2 = - mg\frac{d}{L}\theta(s) \qquad (5)$$

All initial conditions are assumed to be zero

We may determine the transfer function from the gear angle (θ(s)) to the ball position (R(s)) by rearranging equation (5).

$$P(s) = \frac{R(s)}{\theta(s)} = - \frac{mgd}{L(\frac{J}{R^2} + m)}\frac{1}{s^2} \qquad \{\frac{m}{rad}\} \qquad (6)$$

The transfer function is a double integrator. Therefore, it is marginally stable and will provide a challenging control problem.

Section 2.5: Modelling of the System

This section explains the mathematics needed to create the modelling of the system [6].

Section 2.5.1: Mechanical Part: The variables to acquire to understand the modelling process

1. L = Beam length
2. r = Displacement of the ball
3. α(t) = Beam angle of elevation
4. g = Earth's gravitational constant
5. m = Mass of the ball
6. F_{tx} = Translational force
7. F_{rx} = Balls rotational force

The gravity in r direction is

$$F_{tx} = mgsin\alpha(t) \qquad (1)$$

The rotational force of the ball is

$$F_{rx} = \frac{2}{5}mg \qquad (2)$$

If we assume the ball shifts in the direction of r, the contribution of gravity in the r direction will be required to overcome the sphere rotational force in the opposite direction, so assuming the ball displacement is infinite, α(t) is likewise infinite, the linear equation is:

$$sin\,\alpha(t) \approx \alpha(t) \qquad (3)$$

The Laplace transform of the Mechanical Part is

$$\frac{X(s)}{\alpha(s)} = \frac{5g}{7s^2} \qquad (4)$$

Section 2.5.2: The Ball and Beam System Angle Modelling

The angle a(t) between the beam and the horizontal place is achieved by the DC servo motor output in the actual control process, and the relationship between the beam elevation angle a(t) and the motor angle q(t) is non-linear and static, with the gear reduction ratio influencing the relationship. An equation can be approximated in this way.

$$\frac{\alpha(s)}{\theta(s)} = \frac{\alpha(t)}{\theta(t)} = \frac{d}{L} \qquad (5)$$

Section 3: Simulink Modelling:

Building the Simulink model of the ball and beam system consisted of first understanding the mathematical model of the system and then implementing it on to the program. A subsystem was made to take [r, r’, α, α’] and return r’’. The following function is the Lagrangian equation of motion of the ball:

$$0 = (\frac{J}{R^2} + m)r + mgsin\alpha - mr(\alpha)^2 \qquad (1)$$

The function block takes in an input vector u, in this case u[1]= r, u[2]= r’, u[3] = α, u[4] = α’. The mathematical model was then turned into a transfer function; this equation is based off of the Lagrangian model seen above but has been manipulated into a transfer function.

$$\ddot{r} = (-(\frac{1}{J/R^2} + m))(mgsin(\alpha) = mr\dot{a}^2) \qquad (2)$$

Section 3.1: Implementing the PID controller into the system

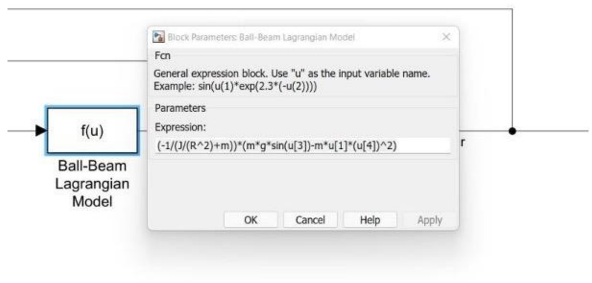


Figure 10

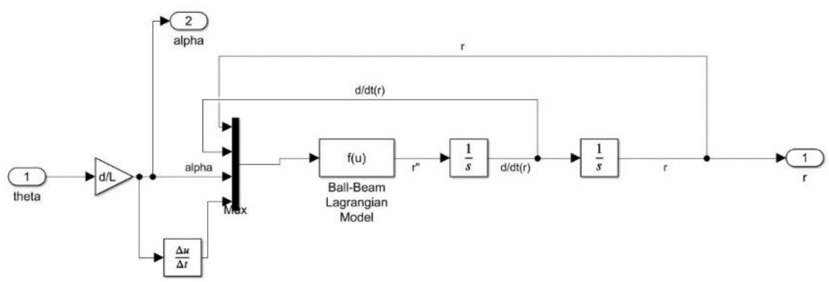


Figure 11

Equation (2) is implemented into the system *(Figure 10)* as you can see the system (Figure 11) takes in theta and outputs a value for r. This system, now is an open loop system and therefore is unstable as a result of this the ball rolls right off the beam. The aim of the ball-and-beam control system is to get the ball to a certain position, r.

<u>Time vs Ball position</u>

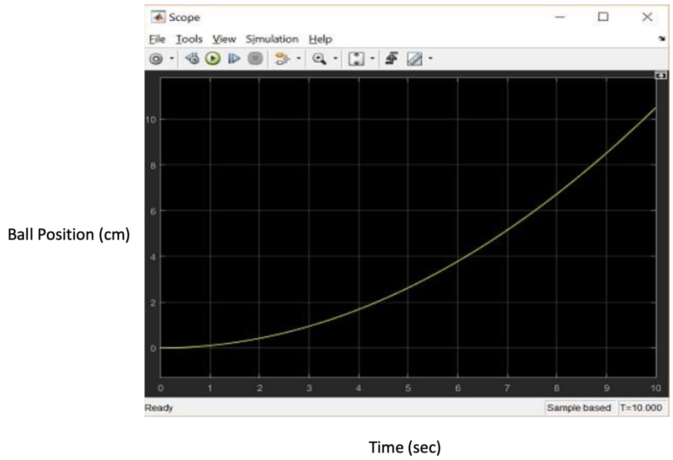


Figure 12

A controller must be implemented so as to control the ball's position on the beam. The chosen controller, a PID was used [7]. The reason for choosing a PID over other controllers was the ease of implementation and feasibility. If a controller is not implemented then it is visible from (*Figure 12)* that the ball will roll off the beam.

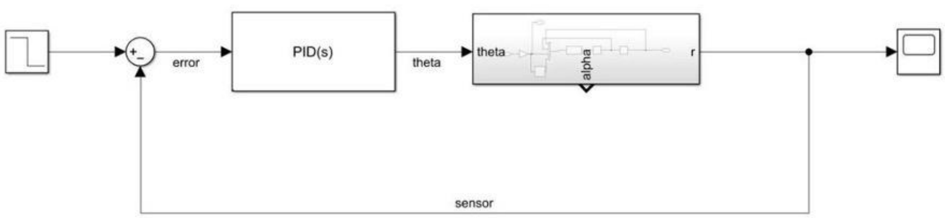


Figure 12

With the PID successfully implemented and a sensor to feedback the information of the balls position on the beam added, the system is now a closed loop system. If we take some arbitrary values and input them into our PID we can see how the system reacts and eventually settles at the desired position of 0.3m

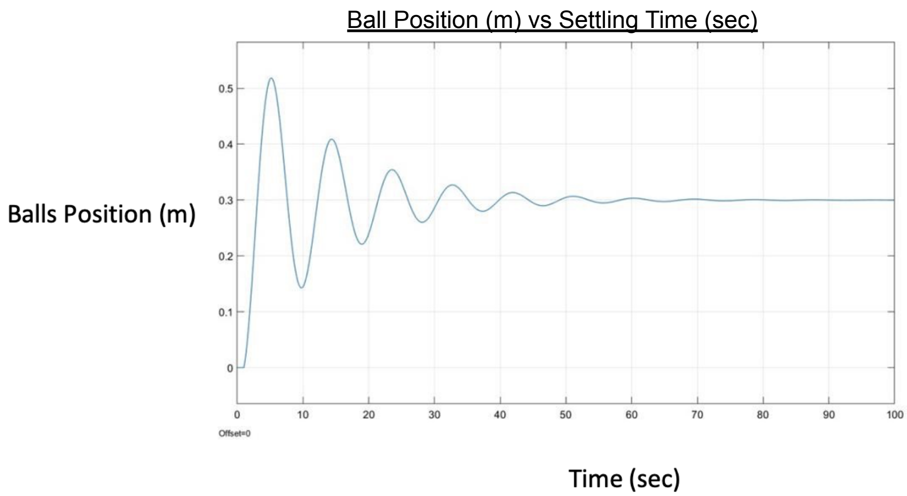
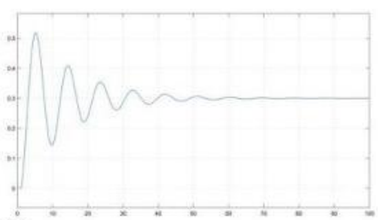
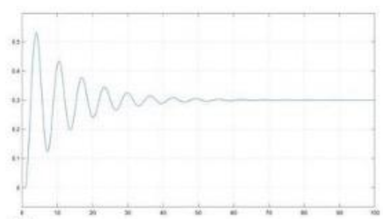


Figure 14

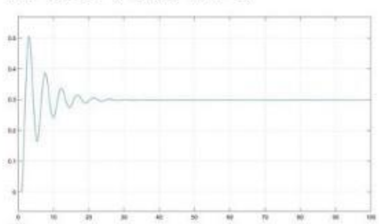
The system now comes to a steady state at the desired position but the settling time is 60+ seconds which can be seen in *Figure 14*. By changing the values on the PID controller the time can be reduced.



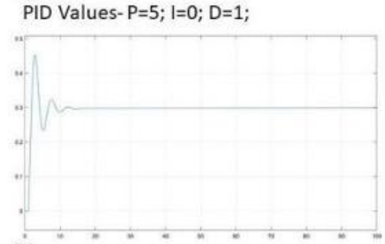
PID Values- P=2.5; I=0; D=1;



PID Values- P=5; I=0; D=1;



PID Values- P=10; I=0; D=2;



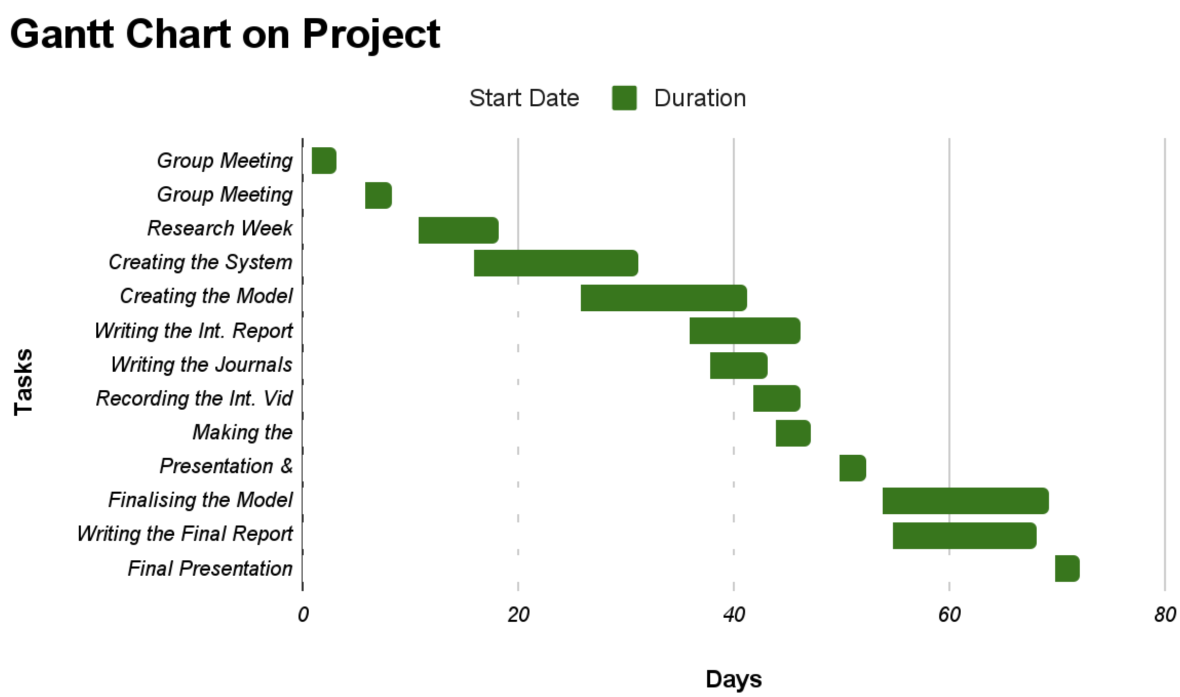
PID Values- P=10; I=0; D=5;

Figure 15

The steady state has now been achieved a lot faster with the last example on *Figure 15* showing the desired steady state achieved in approximately 10 seconds. The values in the PID have been experimented with to find the fastest settling time for the parameters that are given in the project brief.

Conclusion:

The ball-and-beam system is a fascinating subject to investigate. It was captivating to implement the indirect control of the ball location through the control of the beam angle. This project gave the team valuable experience in hardware implementation of digital control. The system was created using the "Wheel and Lever Design". The system consisted of a servo motor, a Arduino, a beam, and an ultrasonic sensor. The mathematical model for this project was derived successfully using the Lagrangian equation. From this we successfully created a Simulink model of the PID controller and the ball and beam itself. From trial and error, the PID values were found to be P = 10, I = 1 and D = 1. This enabled the system to reach a steady state in a shorter period with less overshoot and oscillations. With these findings we can conclude that the system is stable when a feedback control system is used, as shown by the PID controller used.



Future Work:

An aspect that can be improved upon in this project was using a Raspberry Pi instead of an Arduino. Arduino is a microcontroller unlike the Raspberry Pi which is a fully functional computer and system on chip device [8]. The reason for using an Arduino in this project was that it is simpler to both understand and use. It also needs less power than a Raspberry Pi. An advantage to using the Raspberry Pi is that it is more powerful, less knowledge in electronics is needed to be able to use it and it is up to 40 times faster than the Arduino [9].

References:

[1] G. Goodwin, S. Graebe and M. Salgado, *Control System Design*, 1st ed. New Delhi: PHI Learning, 2001, p. 21.
[2] T. Trapp and W. Selby, “Ball and Beam Balance Final Report,” 2009. [Online]. Available: https://s3-us-west-2.amazonaws.com/selbystorage/wp-content/uploads/2016/05/SelbyTrappFInalReport.pdf
[3] Ball and beam centre design - Hauser, J.; Sastry, S.; Kokotovic, P. Nonlinear Control Via Approximate Input-Output-Linearization: The Ball and Beam Example. IEEE Trans. on Automatic Control, 1992, vol 37, no. 3, pp. 392-398
[4] "Control Tutorials for MATLAB and Simulink - Ball & Beam: Simulink Modelling", *Ctms.engin.umich.edu*, 2017. [Online]. Available: http://ctms.engin.umich.edu/CTMS/index.php?example=BallBeam&section=SimulinkModeling. [Accessed: 27- Apr- 2022].
[5] "CTM Example: Ball & Beam Modelling", *Www3.diism.unisi.it*, 1997. [Online]. Available: https://www3.diism.unisi.it/~control/ctm/examples/ball/ball.html. [Accessed: 27- Apr- 2022].
[6] W. Chen, X. Sui and Y. Xing, "Modelling and modulation of nonlinear ball-beam system controller based on matlab," 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, 2012, pp. 2388-2391, doi: 10.1109/FSKD.2012.6234285.
[7] Meenakshipriya, B. and Kalpana, K., 2014. Modelling and Control of Ball and Beam System using Coefficient Diagram Method (CDM) based PID controller. IFAC Proceedings Volumes, 47(1), pp.620-626.
[8] Singh, R. and Gehlot, A., 2020. *Intelligent circuits and systems*. Punjab India: CRC Press, p.359.
[9] Circuitdigest.com. 2022. *Arduino vs Raspberry Pi: Differences between the two*. [online] Available at: <https://circuitdigest.com/article/arduino-vs-raspberryp-pi-difference-between-the-two>; [Accessed 4 May 2022].

[10] "Balancing of a Ball on Beam using Arduino as a PID controller", *Mechatronicstutorials.blogspot.com*, 2022. [Online]. Available: http://mechatronicstutorials.blogspot.com/2014/07/balancing-of-ball-on-beam-using-arduino.html . [Accessed: 05- May- 2022].

Appendices:

Reflective Journals of each team member

Sarah White:

On the surface, I assumed this would be a straightforward assignment, but after doing some research, I realised I was completely wrong! I was pleased to be assigned the roles of mathematics and physics, as I like maths. I started searching up previous reports online to do some study on the mathematical aspect of the project. Once I was happy with my findings I presented them to the group. The group learned later in one of the labs that we would need to revise our design because the one we had chosen was becoming increasingly complex and we couldn't find a relevant report on it. So we went with the "Wheel and Lever Design". I introduced the equations necessary to the group. I started drafting the project's mathematical and physics theory for the report, which would include all of the equations and facts utilised in the maths and physics section. I also created the PowerPoint presentation for the interim presentation and interview stage. After the interim stage, I needed to improve the report and presentation for the final stage of the project and interviews. I added a couple of sections into the final report such as the Final Design section, the Gantt Chart and Appendices section. I also fixed errors that were noted by John Dooley in the interim report. I created a new presentation to reflect the work completed from the end of the interim stage to the final stage of this project along with work completed throughout the entire project.

Tadhg Stones:

Since the interim stage, Eoin and myself have been working on implementing the theory researched in the first half of the project to the actual machine. We had trouble trying to implement the code into the hardware provided, luckily Eoin had a spare Arduino Uno and ultrasonic sensor which seemed more feasible to work with. After a few weeks of trouble shooting we managed to implement the code into the arduino and have the sensor communicate with the servo motor and with further tuning of the PID we got a semi working ball and beam system.

Eoin Keatley:

As part of the ball and beam control system project I was tasked with creating a MATLAB Simulink model of the control system to simulate how the system would work. At this time, I had not used the software before so I would have a steep learning curve. I started by looking at videos and worked examples to try to gain an understanding of how these systems work on the software. I found it hard and decided I would be better off learning the software first, so I went looking and found Simulink and MATLAB onramp courses created by MATLAB themselves. I learned a lot during this about how MATLAB and Simulink work hand in hand to aid you in creating simulations of all types of systems. I learned that in Simulink your goal is to build a mathematical model of your system to show how your system reacts and what values it will emit. I dove into more research about the maths behind the ball and beam control system and found valuable info on equations such as the Lagrangian equation of motion of the ball. This helped towards my goal of creating a Simulink and I managed to make a simulation model of the ball and beam and I could change different values and see how it affected the system. This will be great help for our team as we will have to do much less testing on the physical system as we will already have an in-depth understanding of what values do what and how to optimise the system. I then worked on the finished model where I ran into a lot of issues with the hardware provided to us I had some other hardware at home and used that instead as it is more modernised I was still getting a lot of problems with the hardware as the code was in a language we haven't been taught yet but I managed to understand it enough to get a working model. In future if I was to do this project again, I would try and learn the coding language C much earlier in the project so I could understand it more.

<u>Nathan Ewnetu:</u>
One of the first things we did as a group was meet up and discuss what tasks each one of us should do based on our strengths. I was assigned to code the system. We later discovered that the roles we gave each other were not sufficient to help us reach our goals on time, so we decided to give everyone new roles. The roles given were PID research, different controllers, physics and maths behind the system, MATLAB and Simulink understanding and finally design ideas and displays through CAD programs. I personally help research how PIDs work and how to apply the maths and physics formulas to our ball and beam system. I also worked with Sarah to type up the final report. After understanding the maths, me and Sarah presented it to my team to try and help them understand what I researched. One of the biggest problems I faced during this project was time management. The amount of research our group had to do increased as two of our project members left college. Even though we were short of members we were able to complete almost all the research on time. This showed how we could work together as a team to overcome the problem at hand. Another challenge we faced as a team was in deciding which type of ball and beam design to create. We held a vote and decided on the easier maths design, though we found it very challenging to try and find references and past reports on the design. In the end we proceeded to switch to a design for which we had multiple sources. In my opinion, one of the strengths our team showed was by avoiding conflict through better listening and communication to understand the other person's point of view. I feel that I have benefited greatly from this project. I have also learned it is highly beneficial to have an alternative plan and to plan ahead of time. I thoroughly enjoyed this project-based learning.

<u>Edvinas Ivanovas:</u>
The project that we were assigned to do was the ball and beam. The ball and beam System is one that gives exposure to the basics of a non-linear control system and therefore provides a more in-depth perception of these controllers. The Ball & Beam system is designed to stabilise a spherical object (ball) on top of a beam using control loops using a sensor system which in turn adjusts the torque angle of the motor to provide maximum stability at a particular set point on the beam. This can be done creating and designing a controller in MATLAB SIMULINK. To create this project firstly we needed to research it and know it well by using google as our main source of information, here we were able to find old research papers on this topic giving us all the information needed to complete the first part of the project. To begin the project, we were all separated into our own jobs, mine being to concentrate on gaining information and report writing

whilst also helping the others in any way I could. We booked a study room each Monday to help us connect with one another and view each other's progress over the weekend. I found this to be very beneficial as it began pushing the group to excel in their work having things to show on the Monday's we met. Furthermore, it led us to having more progress during the labs we had on Wednesdays and Thursdays. However, it was not all smooth work as we encountered quite a few problems on the way, the worst being two group members dropping out of the course, therefore making us a group of five instead of seven leading to a bigger workload for each of us. Although being a group of five we managed to keep things on track and go along with our timetable that's set for finishing this project.Throughout the project we learned more about these types of systems and gained more knowledge on each controller. Each member of our group worked in great succession and have helped each other understand the topic they have chosen at the start. Moreover, we finished the project and accomplished the task at hand, and everything worked perfectly.

Code used for the Arduino

[10]

```
#include <Servo.h>
#define Umax 90//ngle of the servomotr in degrees
#define Umin -90//n angle
#define Umax_rad 1.571  // maximum angle of the servomotr in radiants
#define Umin_rad -1.571 // minimun angle
#define T 0.09 // sampling time

const int echoPin1= 6;
const int trigPin1= 7;

int pos;

Servo servo;

double setpoint, setpoint_prec;  // In metres : 30cm --> 0.3m
double y, y_prec;
double error;
double P, I, D, U;
double I_prec=0, U_prec=0, D_prec=0;
boolean Saturation = false;

double Kp =10;
double Ki = 0;
double Kd = 1;
float measure_1 (void);
```

```
void move_servo(int);

void setup() {

  Serial.begin(9600);

  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  servo.attach(9);

}

void loop() {

  setpoint = (.05);  // distance of the cube from the sensor ( metres )
  setpoint = 0.53*setpoint + 0.47*setpoint_prec  ;

  delay(3);

  y = measure_1();  // distance of the cart from the sensor ( metres )
  y =  0.53*y + 0.47*y_prec  ;   // (  alfa*y :   if alfa increases, y less attenuated and
similar to the measured y  --> so the measurement is noisy but fast )


  //Serial.println(y);

  delay (3);

  error = round( 100*(y - setpoint) )*0.01;     // metres

  P = Kp*error;

  if ( ! Saturation )  I = I_prec + T*Ki*error;

  D = (Kd/T)*(y - y_prec);

  D = 0.56*D + 0.44*D_prec;    // filtering D

  U = P + I + round(100*D)*0.01 ;  // U in radiants
```

```
  if ( U < Umin_rad)  {
                  U=Umin_rad;
                  Saturation = true;
                 }

  else if ( U > Umax_rad)  {
                      U=Umax_rad;
                      Saturation = true;
                     }

  else     Saturation = false;

  U=round(U*180/M_PI);     // Transform U in degrees. Now I have :   -63Â° < U < 63Â°

  U=map(U, Umin, Umax, 24, 156); // I map the computed value of U to the
corresponding value of the servomotor

  if (U < 83 || U > 95 || abs(error) > 0.02 ) move_servo( round(U) );   // I continue until I
have an error and the control action U is greater than a threshold.

  delay (24);

  //Serial.print(setpoint*100);
  //Serial.print(" ");
  //Serial.print(y*100);
  //Serial.print(" ");
  // Serial.println(U);
  //Serial.println();

  I_prec = I;
  y_prec = y;
  D_prec = D;
  setpoint_prec = setpoint;
}


float measure_1 (void) {

long durata=0;
float distanza=0;
```

```
digitalWrite(trigPin1, LOW);
delayMicroseconds(10);

digitalWrite(trigPin1, HIGH);
delayMicroseconds(10);

digitalWrite(trigPin1, LOW);

durata = pulseIn(echoPin1, HIGH);
distanza = (float)durata/58.2;

Serial.println(distanza);

delay(30);

if (distanza > 42) distanza=43;
else if (distanza < 0) distanza=0;

return 0.01*(distanza-1.5+0.5);   // metres

}


void move_servo(int u) {
servo.write(u-map(u, 30, 150, 14, 3));

}
```