University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Department of Electrical and Electronic Engineering

University College Cork

VIBRATION REJECTION AND STABILISATION OF A SMART EATING DEVICE

MODULE EE4050 PROJECT – FINAL REPORT

Author: David Moloney
116367623

Project Supervisor: Dr Gordon Lightbody

Date of Submission: 10th April 2020

DECLARATION

This report was written entirely by the author, except where stated otherwise. The source of any material not created by the author has been clearly referenced. The work described in this report was conducted by the author, except where state otherwise.

Signed: David Moloney Date: 10 April 2020

SUMMARY

This report outlines the steps involved in designing and implementing a smart eating device that uses vibration rejection and stabilisation techniques to minimise the effects of tremor seen in the hand of people suffering with Parkinson's disease and essential tremor. The report begins by outlining the problem that is faced by people suffering with these diseases and why electric engineering might solve them. It will then look at the background of some pre-existing solutions to this problem, and what techniques could be used in this project to achieve optimal results.

Two control methods will be used in the project: PID control and phase lead compensation. Each of these will be designed and modelled using Simulink, a simulation software that is an extension of MATLAB. The process involved in designing these controllers and the rest of the control system will be outlined along with the theory behind its functionality. In order to test these controllers a physical system must also be created. This will simulate the movement of tremor in hand of PD and ET sufferers and will give a platform to test the controllers on. The design of this apparatus and the components used within will be explained. The mechanics of this system will be modelled in Simulink as well which will then be used to simulate the system before physical implementation. Following this, the results of these virtual and real-world simulations will be shown and discussed. Finally, conclusions will be drawn from the work carried out and recommendations will be made for future work. The aim of this project is to design an effective stabilisation system that gives equivalent to or better performance than the devices on the market today.

This project was carried out as part of the Electrical and Electronic Engineering bachelor's degree given by University College Cork. It was achieved with the help of fellow electrical and electronic engineering student, Aidan O'Sullivan, who has also provided a report on this project [1].

ACKNOWLEDGEMENTS

This project was made possible with the help of Dr. Gordon Lightbody. A special thanks is given to him and the rest of the staff working in the electrical engineering department in UCC. This project would not have been executed if it wasn't for the help of these people. Thanks to Hillary Mansfield of the control engineering laboratory who helped with acquiring components and implementing the test equipment. Thanks also to Michael O'Shea and Tim power of the mechanical engineering workshop who manufactured the test rig used in the project. The resources that were made available and the continued support from these people was greatly appreciated throughout the course of this project.

CONTENTS

LIST OF FIGURES

2. Background Information

3. Physical Test System

4. Control System Design

5. Results

List of Tables

3. Physical Test System

4. Control System Design

List of Abbreviations

CAGR – Compound annual growth rate

PD – Parkinson’s Disease

ET – Essential Tremor

IEEE - Institute of Electrical and Electronics Engineers

PID – Proportional-Integral-Derivative

PI – Proportional-Integral

PD – Proportional-Derivative

ADRC – Active Disturbance Rejection Control

RHS – Right Hand Side

LHS – Left Hand Side

COG – Centre of Gravity

DC – Direct Current

IMU – Inertial Measurement Unit

PWM – Pulse Width Modulation

EMF – Electromotive Force

IDE – Integrated Development Environment

I/O – Input/Output

MEM – Micro-electro-mechanical

I2C – Inter-integrated Circuit

DOF – Degree of Freedom

CW – Clockwise

CCW – Counter-clockwise

List of Scientific Symbols

K_P – Proportional gain

K_I – Integral gain

K_D – Derivative gain

α – Estimator Proportion gain

β – Integral estimator gain

θ – Platform angle

φ – Spoon angle

e - Error

R_P – Platform setpoint

R_S – Spoon setpoint

J_h - Platform moment of inertia

J_s – Spoon moment of inertia

K_m – Motor torque constant

K_m – Motor back-EMF constant

N – Gearbox ratio

R – Terminal resistance

M - Mass of platform

m – Mass placed on spoon

g – Acceleration due to gravity

d – Distance to platform COG form platform axle

l – Distance from mass on spoon to spoon axle

ω_p – Pass frequency

ω_s – Stop frequency

ω_c – Corner frequency

n – Order number

K_u – *Ultimate gain*

T_u – Period at ultimate gain

T_s – Sample time

A_Y – *Acceleration in Y-direction*

A_z – *Acceleration in Z-direction*

1. INTRODUCTION

Technology today continues to expand and progress within many industries. One industry that is making massive advances in technology innovations, is the medical sector. The global market for medical devices is expected to reach $640.9 billion by 2023 at a compound annual growth rate (CAGR) of 6.4% [2]. Micro-electronics and computers have many uses within the medical sector including surgical instrumentation, imaging and data collection [3]. One factor that is driving this market is the rising demand for innovative therapies to overcome unmet needs in the healthcare system. Where pharmacological and surgical intervention are less effective in the treatment of certain diseases and conditions, electronic solutions may hold the answers. One such condition is tremor.

Tremor refers to involuntary movement that is seen in the body. The condition can show for a number of reasons that can be either physiological or pathological [4]. Physiological tremor is inherent in all people and is often unnoticeable. Pathological tremor is much more invasive and can have devastating effects for people who encounter it. Pathological tremor can become present for a number of reasons, one of which is due to Parkinson's disease (PD), the second most common form of pathological tremor [5]. PD is a neurological disorder is caused by the lowering of dopamine levels within the brain and nervous system [6]. It is neurodegenerative disease that is triggered by the death of dopaminergic neurons. The resultant dopamine deficiency leads to a movement disorder that is characterised by involuntary movements and vibrations in the human body, with a frequency typically in the range of 3-7 Hz [5][7]. Although it is not the most common movement disorder it is widely known because of severity in patients and the difficulty it brings after its onset. The debilitating nature of PD has been known for centuries and with about 1 in 500 people contracting the disease during their lifetime, treatment of the condition is often a point of interest [8]. The effects of PD are extremely impairing for sufferers and will prevent them from performing basic tasks that every human requires for a good quality of life. Carrying out such tasks as eating, dressing and even walking are greatly impeded as the disease progresses in patients. While the physical hardship is awful for people with PD, so too is the emotional. As PD becomes more serious in patients, their independence in life diminishes, which can have serious effects on one's mental state. Loss of dignity and freedom can lead to depression in up to 45% of people with this devastating disease [9].

The most common movement disorder today is Essential tremor (ET). This is similar to PD in that it is a pathological disorder that leads to involuntary vibrations in the body, with frequencies of about 4-12 Hz. In this case the symptoms are not as debilitating, as tremor is usually seen in just the hands [10]. Resting and postural tremors are not seen in the same ferocity as that in PD but kinetic tremor remains an issue, that is tremor seen when intentionally moving the body [5]. Like PD, the presence of ET can mean that carrying out basic tasks can become a large obstacle leading once again to lower quality of life and lack of independence.

As the root cause of PD and ET is not yet known, a cure for these diseases has not been found. Treatment today comprises of medications that either replace dopamine or mimic the effects of the neurotransmitter, as well as deep brain stimulants and rehabilitation strategies [11]. These have shown to be somewhat effective but do not give patients a normal quality of life. Moreover, the effects of medications often wear off as the disease worsens [5]. For this reason, attention has now turned to technological solutions that can improve the standard of living for PD or ET sufferers [12].

As feeding oneself is such a critical part in one's life, the need for assistance with such a task can be very disparaging. To aid people who experience tremor, a device may be devised that can look to remove the effects of tremor and thus give back independence to the people who suffer with it. One option is to utilize a form of exoskeleton like that created by Steadiware [13]. This device makes use of smart fluid and a mass dampener which takes inspiration from earthquake protection technology used in skyscrapers. This performs well but can be quite intrusive and may be humiliating to wear at times. Applying a more subtle approach leads to the idea of creating utensil that can remove the effects. This project aims to explore such a solution in the designing of a smart eating device that uses vibration rejection and stabilisation techniques to remove the effects of tremors seen in the hand of people with PD or ET. The "smart-spoon" may even be helpful for people suffering with other movement disorders such as Huntington's disease and multiple sclerosis. The device aims to re-establish dignity and self-reliance thus improving the quality of life of these people.

This project began with research looking at pre-existing solutions to similar problems and the control engineering techniques that might be implemented to achieve the task. Different forms of controllers were investigated along with signal processing methods. Following this a means of testing the system was devised that would mimic the movement of the user and the device itself. Once this was established, the system mechanics were modelled, and a control system was designed and implemented in software. Finally, results were gathered, and conclusions were drawn for the basis of future work.

2. Background Information

2.1. Pre-existing Solutions

Two similar devices currently sit in the market that provide a solution to this problem. These are the Liftware Steady by Lift Labs, and the GYENNO spoon by GYENNO Science, both pictured below in figures 2.1 and 2.2 [14][15]. Both devices are designed for people who experience tremor and more specifically those who have PD. Both aim to remove the effects of tremor by measuring movement in the hand and counteracting the unintentional vibrations by driving the spoon in the opposite direction. An ergonomic design is used in both products that consists of rounded body and spoon head that is moved in two dimensions using actuators within the device

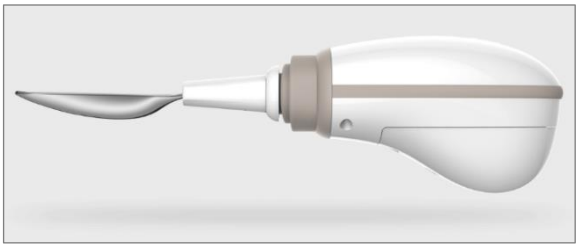


Figure 2.1 – GYENNO Spoon [15]

Figure 2.2 – Liftware Steady Spoon [14]

The GYENNO spoon makes use of two high speed servo motors that give 360° stabilisation. A proportional-integral-derivative (PID) control algorithm is used in the device, along with an attitude stabilisation technology like that seen in unmanned aircrafts [15]. The result of this is excellent control performance and high frequency response. The device uses sensors that measure the motion of the hand and distinguishes between the involuntary tremors associated with PD and the intentional movement that is required to eat with the spoon. The Liftware Spoon employs a similar method of solving the problem. Again, it measures the movement of the hand and actuates the head of the spoon along two dimensions in order to compensate for the tremors. The custom control algorithm implemented in the device ensures that the head of the spoon follows the intentional movement of the user while disregarding the tremor. Both devices achieve good results with tremor attenuation levels of 85% and 70% boasted by the GYENNO and Liftware devices respectively [14][15]. While these products offer clean, modern form factors with added functionality, this project did not look to achieve such results. The goal of this project was to design a control system that can be implemented in a similar device that will achieve better or equal levels of vibration reduction performance.

Founder and CEO of Lift Labs, Anupam Pathak, took inspiration from his work with Army Research Laboratories where he was designing stabilisation systems for weapons to be used in combat [16]. Considering this, a solution may be found be investigating other stabilisation technologies. One such area is that of camera stabilization. Ali Algoz and Bakhtiyar Asef Hasnain of Uppsala University looked at designing a control system for a 3-axis camera that could account for sudden movement in the person holding the device [17]. While our project will only operate in one dimension, the controller design is similar. They achieved reasonably good performance from their PID controlled system. They did, however, run into some issues with their brushless DC motor, in that some torque cogging was observed when the motor was operating at low speeds. This was a critical aspect to consider when designing the physical system of this project. A similar project run by two professors in VIT University also looked at designing a PID controller for 3-axis camera stabilization. The pair achieved a high level of performance by using complex tuning methods like particle swarm optimization and genetic algorithm tuning [18]. Finally, one of the more well-known camera stabilization technologies comes from Steadicam which uses an iso-elastic arm attached to a low friction multi-axis gimbal that holds the camera at one end and a counterbalancing weight on the other [19]. Although this offers excellent stabilisation, it may not be the solution for this problem, as it uses mostly physical mechanisms, rather than an electronic controller.

While camera stabilization techniques may be the answer to this problem, other devices may also hold the solution. A 2013 paper published by IEEE, outlined the design of a surgical instrument that compensated for physiological tremor in the user's hand [20]. While camera stabilization looks at the problem on larger scale, this seeks to solve the problem on a much smaller scale, removing the millimetre sized tremors present in the hand. The device senses the motion of the user, filters the high frequency tremulous motion from the lower frequency intentional motion and deflects the tip to compensate for any involuntary tremor.

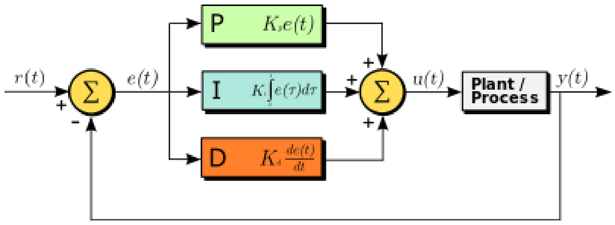
Finally, the work that gave the best guide for completing this project is that of Rob Shanahan and Emily Long of University College Cork [21]. In 2019, they looked at vibration suppression in a "Smart Spoon" using active disturbance rejection control (ADRC), a project that we have decided to build upon and hopefully improve. The pair used a spoon and platform system which simulated the "smart-spoon" device along with the movement of a tremoring hand. They decided to use the ADRC and linear ADRC methods of control. Due to issues with their motor they only achieved moderate results with their device. As well as this, in attempting to use ADRC and LADRC, they were not able to fully implement their system. These issues would both be taken into account over the course of completing this project.

2.2. Control Methods

The following section outlines some of the control methods that could be used to ensure that the spoon tracks the intentional motion of the user.

2.2.1. PID Control

Within a closed loop control system, the controller algorithm is used to convert an error signal to an actuator signal. The error term is generated by comparing the setpoint signal to the feedback sensor signal. Proportional-integral-derivative (PID) controllers are the most common controllers used in industries due to their effectiveness and simplicity. The PID controller applies proportional, integral and derivative functions to the error signal which are then summed together. The block diagram of a closed loop control system using this form of controller is shown below along with the transfer function describing the controller.



$$G_{PID}(s) = K_P + \frac{K_I}{s} + K_D s$$

Figure 2.3 – PID Controller Block Diagram [22]

The proportional part of the controller corrects the error by applying a correction that is proportional to the error. For this reason, the set point is never achieved, due to the fact that as the difference approaches zero so too does the correction. The integral part of the transfer function sums the error component over time. The integral response will continue to increase as long as an error is still present in the system. The result of this is that it drives any steady state errors that are present in the system, to zero, but also means that an overshoot will often be seen as it drives the output towards the setpoint. To account for this overshoot, the derivative component is used. The derivative part of the transfer function looks at the rate of change of the error and accounts for the expected error that will become present as it overshoots the set point [23]. The performance of PID controller is largely dependent on the magnitude of each of the gains used. These can be tuned individually until the level of control desired is obtained. For this reason, the controller is very versatile and finds use in many automation and control systems.

2.2.2. Phase Lead Compensator

To understand the effect of a phase lead compensator, it is worth understanding the plotting of transfer function poles and zeros in the s-domain and root locus diagrams. These will be used later in the design process, so it is useful to explain these concepts at this stage. Taking the Laplace transform of a time domain signal yields the transfer function of a system and allows one to manipulate the function in the s-domain. S is a complex number in the form $\sigma + j\omega$, which can be plotted on a cartesian plane called the s-plane. The roots of the characteristic equation of a system can show certain performance characteristics which are found by equating the transfer function to zero. These are known as closed loop poles. Plotting them in the s-domain gives an insight into how the system performs showing the stability, damping factor and natural frequency. Each root location on the s-plane represents a waveform in the form e^{st} and the presence of multiple locations will give some combination of these waveforms [24]. The graph below shows how varying the location of these gives different impulse response.

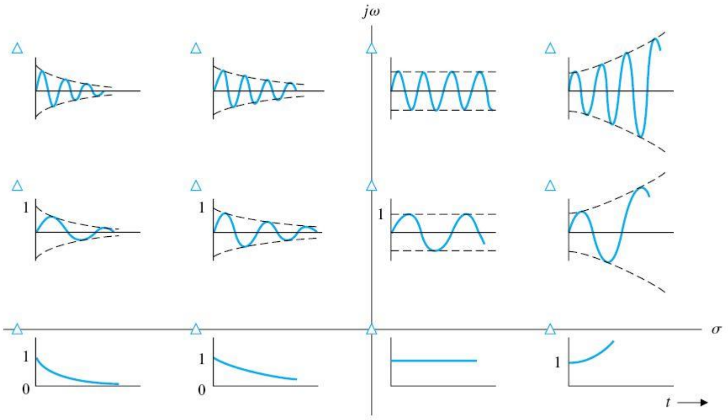


Figure 2.4 – Impulse response at varying poles locations [25]

Values of s that are purely real give exponential behaviour be it positive or negative. As the position is moved further from the origin the rate of growth or decay increases. Closed loop pole locations that have a purely imaginary value give sinusoidal behaviour with an increase in frequency as the imaginary value increases. A value of s that is comprised of real and imaginary parts will give a combination of these, with exponential and sinusoidal behaviour. It should be noted that the presence of any complex pole locations means that a pole is also located at its complex conjugate. The concept of stability becomes clear here by seeing that any locations in the left side of the plane gives a system that returns to zero when an impulse is applied. In the case of a control system, the transfer function may be made up of the controller/compensator and the plant system. If the system is stable a change to the error input will return to zero after some time, thus bringing the output of the system to the setpoint.

If there is an unknown in the transfer function, then the closed loop poles will change by varying this unknown. If every possible value for said unknown was used and each of the set of poles were plotted on the s-domain then the resulting graphs would be a line that runs through every one of these points. This type of plot is known as a root locus diagram, an example of which is shown on the next page in figure 2.5. In this case, the unknown is a gain, k, and increasing it will take the closed loop poles to $\pm\infty$ along an asymptote at -2.5.

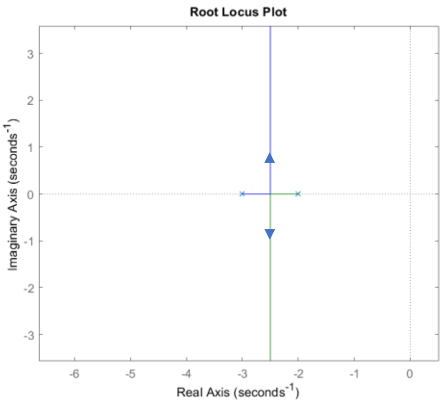


Figure 2.5 – Root locus diagram

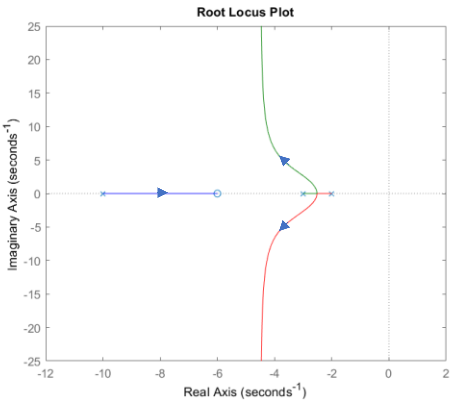


Figure 2.6 – Root locus diagram with compensator

Figure 2.6 shows the effect of adding a phase lead compensator made up of one pole and one zero, in which the pole location is further from the origin. The asymptote has now been moved, with the effect of bringing the root locus further into the left-hand plane. A gain can then be chosen to achieve closed loop poles at some location along this plot. Stability can be added to an unstable system by pulling the root locus from the RHS to the LHS of the s-plane. The addition of a phase lead compensator will add another closed loop pole but the effect of this is not seen as the poles closest to the origin dominate the behaviour. These are known as dominant closed loop poles [24]. This method can be used to create a stable system that brings an impulse in the form of an error signal, to zero.

2.3. Project Description and Objectives

After carrying out research for this project, a high-level design was conceptualised, and a plan was laid out to achieve the goals of this project. The basic premise of our design will be similar to that used in Ref. [11]. The device will consist of two sensors, a filtering system and actuator in the form of a motor. The first sensor will record the movement of the hand of the user. This will then be passed through a filter removing the tremulous frequency component. The filtered signal will then become the setpoint for a closed loop control system that uses readings from a second motion sensor which is placed on the spoon. As the tremor is removed from the setpoint signal, the spoon will only follow the intentional motion of the user.

The following chapter will look at the physical system that was designed and created to replicate the movement of a tremoring hand along with the spoon mechanism actuated by the control system. In chapter 4, the control system design will be outlined. Here the mechanics of the system will be modelled, a filter will be designed, and a controller algorithm will be developed. Following this the results of both the simulations and real-world tests will be explained and finally, the project as whole, and what can be taken from it, will be discussed.

3. Physical Test System

3.1 System Design

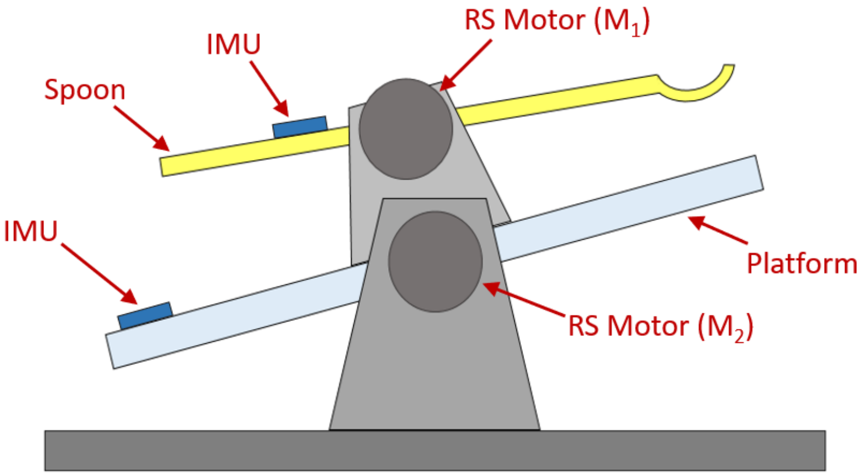


Figure 3.1 – Physical System Diagram

In order to test the performance of the controller algorithms designed, a physical test system had to be devised. The design chosen was similar to that of a previous project carried out by Rob Shanahan and Emily Long [21], with a few alterations made to improve the reliability of the tests carried out. The diagram above shows the test apparatus that was designed. Here it can be seen that the testing apparatus is made up of two primary sections: the platform and the spoon. The platform is used to mimic the handle of the device and thus the hand of the person using it. This platform is free to rotate about an axle running horizontally through its centre of gravity (COG). This axis is supported on both sides by aluminium brackets seen as trapezoids in the diagram above. A high-quality axle was used which showed little friction when moving and did not introduce any undesired mechanical vibrations into the system. The platform was moved by an DC motor, labelled above as RS motor M_2. This motor was used to give replicable tests, mimicking the hand of a person with PD. This was one of two actuators used in the physical system. The other motor, M_1, was identical and was used to move the second the half of the system, the spoon. As one might expect, this was used to represent the spoon head in our device. Again, this was free to rotate about a horizontal axle through its COG and was supported by two more brackets which were then mounted to the platform. This symbolised the spoon that would be mounted within the handle of the device which would then be moved using some form of actuators.

Two forms of sensors were used in this system. The first of these were inertial measurement units (IMUs). These would be used for positional tracking, and more specifically angular displacement measurement. Two of these would feature in the smart spoon device, one in the handle (platform) and one on the spoon. Both sensors would be used for controlling the spoons movement and removing vibrations due to tremor. The second type of sensor was a potentiometer. Two of these were again used, one for the spoon and one for the platform. Both were mounted on the opposite side of each axle and would also be used for measuring position, this time for checking the results of tests.

Care was taken in selecting the material to be used in the test apparatus. Most importantly, a low-density material was used in making the spoon. The purpose of this was to create a low moment of inertia, which meant the movement of the spoon would not have as much impact on the platform. It also gave a more accurate representation of the hand-spoon mechanics.

3.2 Physical System Components

3.2.1 Motors

The two motors used in this system were brushed DC motors manufactured by Portescap. Both were identical in size and rating, with a nominal voltage of 6V and a maximum output torque of 8.4 mNm. The brushed DC motor works on simple principle. When current flows through the device, opposing magnetic fields are generated in electro-magnets attached to the motor's armature. These are then either attracted or repelled by permanent magnets in the stator, causing the armature to spin. Once the armature has reached its point of equilibrium within the magnetic field, the current within the armature is reversed, thus flipping the direction of the magnetic fields in the electromagnets. This continues to occur at each equilibrium position meaning the armature continues to spin with an angular velocity that is proportional to the current flowing in the device. The "brushed" term refers to the method by which the current is supplied to the windings. Each time the current is switched it is due to brush connections that are passing over oppositely charged terminals.

As the resistance of the wire within the armatures is considered to be constant, the current flowing can be controlled by varying the applied voltage. Rather than applying a constant DC voltage to the motor that is somewhere between 0 and 6 V, a pulse width modulated (PWM) signal is used instead. Changing the duty cycle of square wave signal that oscillates from 0 and 6 V will give an average voltage that is somewhere in this range. The average voltage seen at motor is directly proportional to the duty cycle of PWM signal.

The Portescap motors used in our experiment were chosen for their superior quality. These motors were made with great precision which means high accuracy and less mechanical complications. A problem that was seen in experiments of Ref. [21] was a deadband in their system due to backlash in their motor. Backlash refers to presence of space between mechanical gears in a motor thus leading to delay and inconsistencies in output torque. This meant the performance of their control system was massively hindered as their spoon was not being reliably moved by their motor. A deadband may also be present due to stiction within a motor. While some of these issues can be aided using software, any imprecision seen within the device can lead to mechanical vibrations and noise when it is used as part of a larger system. For these reasons, more expensive but higher quality motors were chosen.

The motors used were repurposed, having been utilised in another project at an earlier stage. Both motors were connected to a 40:1 gear box. This meant the output torque and speed of the motor were both altered by a factor of 40. The output torque was multiplied by 40 giving maximum value of 336 mNm and the speed was divided giving a maximum value of 300 rpm (5 Hz). A calculation was carried out to ensure that the motor could reach the speed required for our experiments. Considering the fact that the angle is changing in a sinusoidal fashion, then the platform angle, θ, can be given as.

$$\theta(t) = Asin(\omega t)$$

Where A is the amplitude of the signal and ω is the frequency of the oscillation. In order to replicate tremors, it was decided that 0.3 rad and 18.8 rads^{-1} (3Hz) would be used for these values respectively. Taking the first derivative of this gives the angular speed:

$$\dot{\theta}(t) = A\omega cos(\omega t)$$

The maximum angular speed is then $A\omega$ rads^{-1}. This can then be converted to frequency as follows:

$$\frac{A\omega}{2\pi} = \frac{0.3(18.8)}{2\pi} = 0.898 \text{ Hz}$$

With a maximum speed of 5 Hz, it is clear that the motors used can generate sufficient speed to be used in this project.

3.2.2 Motor Shield

The Arduino motor shield was used as the interface between the motor and controller of our system. As seen in the figure here (right), the motor shield is similar in size to an Arduino Uno. This meant it could be placed on top of the Uno and still give access to each of the I/O ports. The device uses a dual full bridge driver to control two forms of solenoid or actuator. In this case

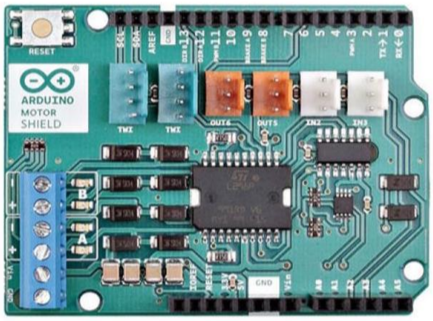


Figure 3.2 – Arduino Motor Shield [26]

the direction and speed of each of the motors is controlled. As can be seen in figure 3.2 , the two motors are powered through connections at the bottom left corner of the device. A larger DC supply can also be added here through a third pair of connections. This would be necessary if the operating voltage of the motor was higher than the 5 V supplied by the Arduino Uno. In the case of this experiment this will not be required. Digital outputs on the microcontroller were used to control the motors. Each channel required two digital outputs, one of which dictated direction (clockwise or anticlockwise) and the other controlled speed using a PWM signal. The duty cycle of this signal was dictated using an 8-bit value set in the Arduino Integrated Development Environment (IDE).

3.2.3 Arduino Uno

As previously stated, the microcontroller used in the experiment was an Arduino Uno Rev. 3, seen in figure 3.3. This was chosen as it gave a huge amount of functionality and was straightforward to implement. Within the group there was already some experience working with the device, so it was the natural choice. The board uses an ATmega328P microchip from Atmel. It operates at 5 V and offers 14 digital I/O pins (6 of which provide PWM output) and 6 more analog input ports. Although the Uno is larger in size than other controllers it gives a great initial test bench for designing our system. Its open source design means that once the functionality is decided upon, a new board can be printed, removing any unnecessary parts. The key specifications of this device are outline in table 3.1 below.

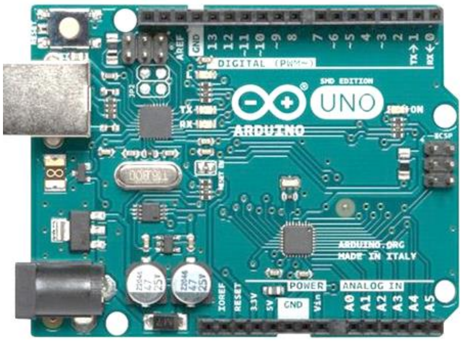


Figure 3.3 – Arduino Uno Rev. 3 [27]

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Digital I/O Pins	14
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Table 3.1 – Arduino Uno Specifications

The Arduino controller was powered through a USB port connected to a desktop computer. This ensured the input voltage was kept within its limits of 7 and 12 V, which would otherwise lead to inaccuracy in output voltage settings and input voltage measurements. This might also cause the voltage regulator to overheat which can damage the board. The USB connection was also used for transferring information to the board. Scripts were coded in C++ coding language on the Arduino IDE where libraries could be added allowing functionality of 3rd party peripherals. The code was then uploaded to the Arduino Uno where it would run through the script outlined.

3.2.4 Potentiometers

Two rotary potentiometers were used in the testing system to accurately measure the angular displacements of the platform and spoon. These were attached to the axles that the spoon and platform were mounted on. The rotary potentiometer works by sliding a wiper contact across a resistive element. The resistor is connected on one side to a supply voltage and on the other to ground. The voltage on the wiper is then measured and a variable voltage divider is created. The wiper voltage was measured using one of the analog input ports on the Arduino Uno. This converted a voltage between 0 and 5 V to an integer between 0 and 1023 (10-bits). Using the voltage sources supplied in the laboratory, a voltage was applied to the high terminal. This was set to a value so as not to exceed 5 V at the wiper terminal when the platform/spoon were at their maximum angles. In both cases the voltage range seen at the wiper was around 2.55 V. This could have been increased to 5 V using a pull-up circuit, but it was decided that the level of resolution achieved by the 10-bit analog inputs was sufficient for graphing results.

3.2.5 IMUs

IMUs are used in many applications for position and movement tracking. The device chosen to be used in this experiment was the Sparkfun 9DOF Sensor stick featuring a LSM9DS1 chip. These consist of numerous micro-electro-mechanical (MEM) devices which are used in an accelerometer, gyroscope and magnetometer, each measuring along 3 axes (9 degrees of freedom). For the purpose of this experiment only two of these sensors are used in each of the IMUs; the accelerometer and gyroscope.

The accelerometer measures acceleration using a MEM spring and mass system. When the device accelerates a mass connected to a spring moves within the device. According to Hook's law the displacement of the mass is proportional to the force exerted on it. The displacement can be found by measuring the change in capacitance due to fingers on the moving mass. Knowing the mass of the object and the spring constant, the acceleration of the device can then be found.

The gyroscope uses a measurement technique which is less trivial. It features a MEM Coriolis vibratory gyroscope. The principle that this operates on is the fact that a vibrating object will continue to travel in the same direction when it is being rotated. By considering the effect of Coriolis acceleration the speed of angular displacement can be measured. The diagram below shows how the resonating mass reacts to a spinning motion. The average speed of the moving mass is that at the centre point of its path of oscillation. When it moves towards the outside of the rotating body the mass is now travelling through space at a slower rate than the body around it, causing it to accelerate to the right. The mass then applies a reactive force left which can be measured. When it moves towards the centre of the rotating body, the mass is moving faster than the surrounding structure and therefore exerts and reactive force to right [28]. The forces generated by the moving mass are measured using similar techniques to the acceleration and with this the angular velocity and direction of rotation can be derived.

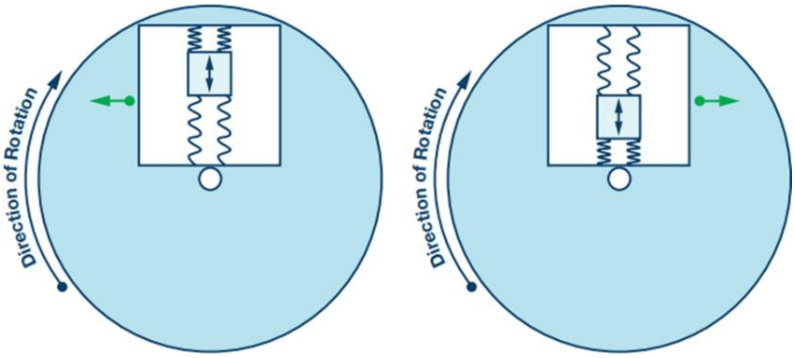


Figure 3.4 – MEM Gyroscope Diagram [28]

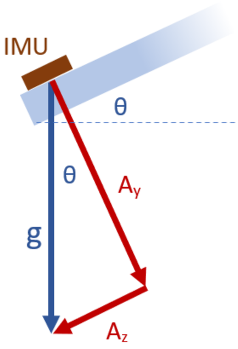
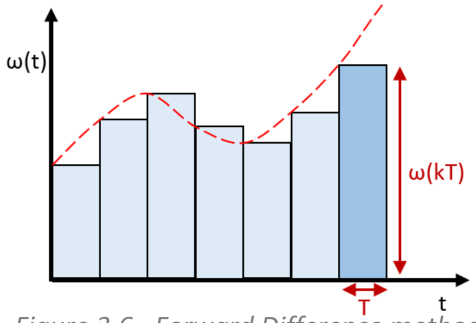


Figure 3.5 – Acceleration seen by IMU

Angular displacement was measured using two methods regarding the IMU. One method used two of the acceleration readings, both which were placed along the plane of rotation. Figure 3.5 shows the IMU attached to the platform. The devices see an acceleration due to gravity which is measured in part by the two accelerometers. Finding the arctangent function of the ratio of A_z to A_y will give the angle at which the platform/spoon is orientated.

As the gyroscope measures the angular velocity of the system, then the angular displacement can be found by taking the integral of this value. To achieve this, Euler's forward difference method of digital integration was applied [29]. Integration can be approximated by finding the area under the curve. This area can continuously be found by adding on the area between two samples. This can be viewed as a rectangle with height equal to the value of the sample and width equal to the time step. This can be seen more clearly in figure 3.6. Applying this method, a general equation for the angle can be created. This equation can be implemented on the Arduino code and thus the angle can be calculated.



$$\theta(kT) = \int \omega\, dt = \theta\big((k-1)T\big) + (\omega(kT) \times T)$$

Figure 3.6 –Forward Difference method

While both of these methods may be used to measure the angle of the platform, they each have their own inherent problems. The accelerometer method is susceptible to noise and gives inaccurate values at high oscillation frequencies, as the IMU is experiencing acceleration other than that due to gravity. The gyroscope method then runs into its own issues due to drift seen at stationary positions. This is due to area under the curve that is missed by the approximation above. The solution to this is to use both measurements and combine them together using some sort of filter. This could be in the form of Kalman or a complimentary filter that removes high frequency accelerometer signals and low frequency gyroscope signals. The solution that was decided upon, however, was to use a first order estimator, that took some portion of each signal which together made up an accurate angle measurement. A block diagram of this method is shown below.

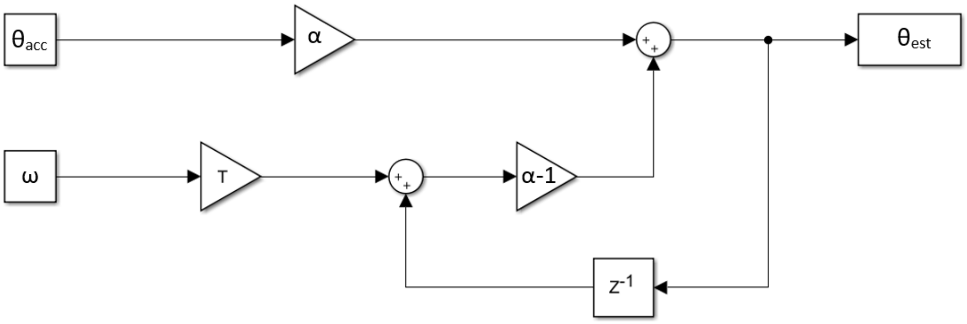


Figure 3.7 –First Order Estimator Block Diagram

The equation for the estimated angle, $\hat{\theta}$, in terms of sample, k, is then:

$$\hat{\theta}(k) = \alpha\theta_{acc} + (\alpha - 1)(\theta(k-1) + \beta(\omega(k)T))$$

Another gain, β, is also introduced here. This was used to improve the performance of the integrated angular velocity value. This was included to account for errors in the sample time used and in the integration estimation explained above. This estimator then had to be tuned by adjusting the two values of α and β until the estimator gave accurate readings. This was achieved by measuring the angle of the spoon/platform with their respective potentiometer and recording values for both accelerometer and gyroscope reading at the same time. All values were then imported to MATLAB where the estimator was implemented. Plotting the potentiometer and IMU reading gave an insight into how changing the gain values effected the angle measurement generated. α and β were then adjusted until both plots matched each other.

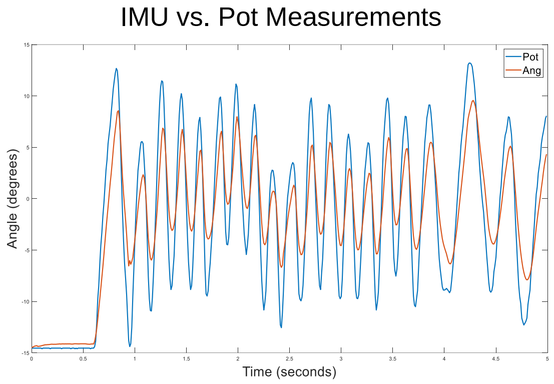


Figure 3.8 – Initial IMU vs. Pot graph

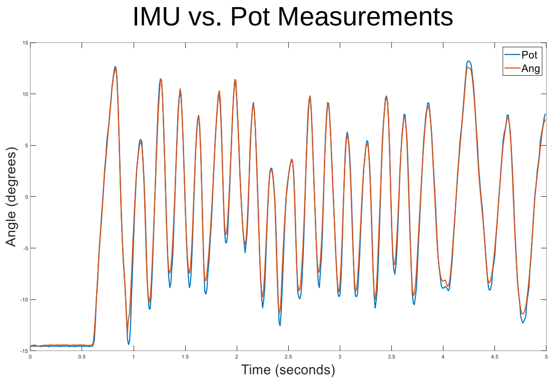


Figure 3.9 – Final IMU vs. Pot graph

Figures 3.8 and 3.9 show the effects of tuning the estimator. The final values were 0.02 and 0.95 for α and β respectively. The small value for α found meant that the drift in the gyroscope readings was compensated for by applying a small portion of the accelerometer readings. This meant an error did not build up over time and the angle reading was accurate under all conditions.

3.3 Finalised System

3.3.1 Physical System

The final testing system used can be seen in figure 3.10, on the next page. Here the spoon and platform can be seen clearly. Two stops were added that were used to hold the platform in place and prevent it from moving too much. The base of the device and the brackets were made of aluminium 6082. A light plastic, called ABS plus, was used in the spoon which was 3D printed and the platform was made of a clear polycarbonate. The two DC motors can be seen at the near side of system while the two potentiometers are on the far side. One IMU can also be seen on the platform and the other was placed at the centre of the spoon.

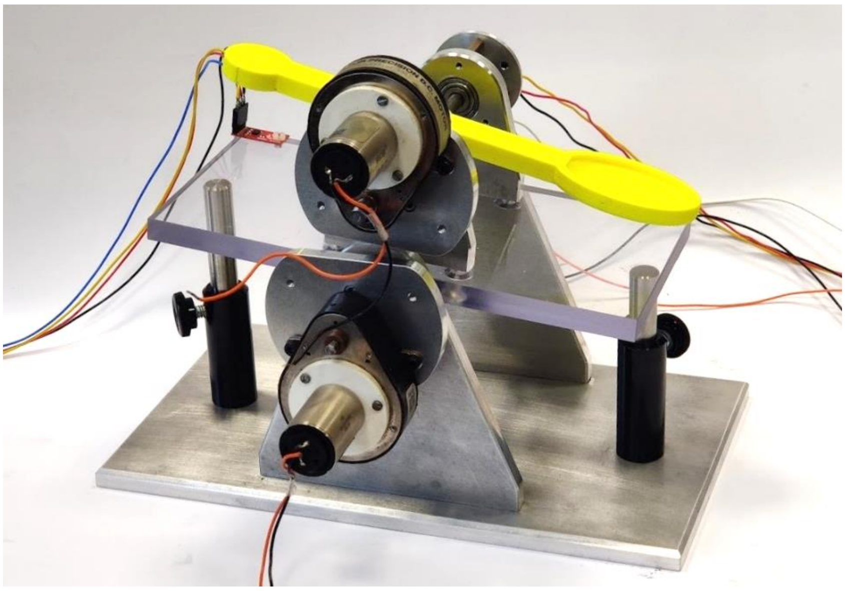


Figure 3.10 – Final Physical System Used

3.3.2 Circuit Diagram

Figure 3.11 outlines the circuit diagram of the test system. In the actual set up the motor shield was placed on top of the Arduino Uno so connections to ports 8, 9, 12 and 13 were already made and the rest of connections to the Arduino were made through the motor shield board. The two IMUs can be seen in red. Only four connections are required for each of these as they use I2C communications. Both devices transferred information through the same clock and data connections seen as yellow and blue below. As well as these, there are the two motors powered through the motor shield and the potentiometers with supply voltage of approximately 30 V.

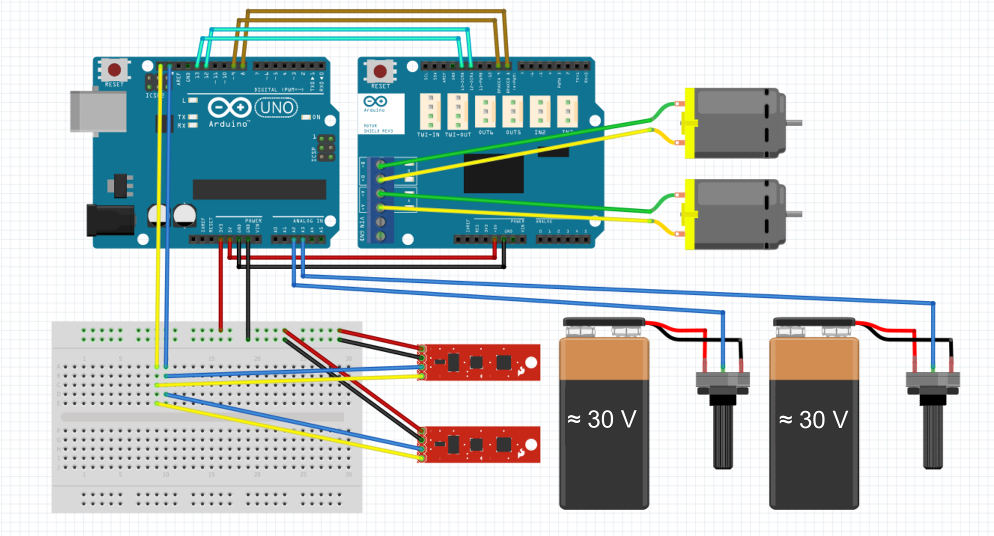


Figure 3.11 – Circuit Diagram of Testing System

4. Control System Design

4.1. Control System Overview

The following section explains the high-level overview of the control system. A block diagram of this design is shown in figure 4.1. After designing the testing apparatus, it was noticed that the weight of the spoon motor meant that the platform system was unstable and if it were to be moved using a motor a second control system would be required. This can be seen in top half of the block diagram below.

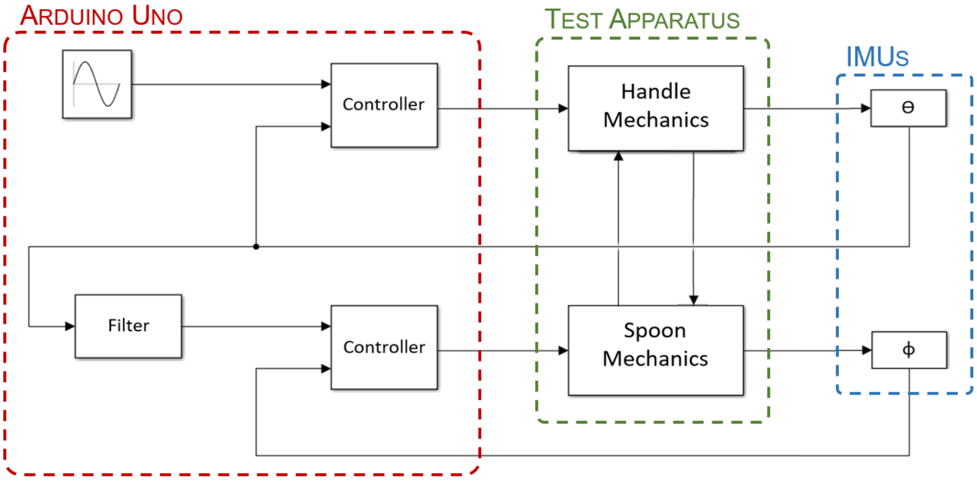


Figure 4.1 – Control System Overview

An initial setpoint for the platform movement is generated using software. This signal is made up of a high frequency component represent tremor movement in the hand of a user and a low frequency component representing intentional movement. It was decided that a 3 Hz (18.8 rads^{-1}) tremor frequency would be used as it was the lowest frequency seen in both PD and ET sufferers and would therefore be the hardest to effectively remove with the filter. This signal is then passed into the first control system which is used to move the platform. This generates a voltage in the form and PWM signal that drives a motor dictating both speed and direction. The angular displacement of the platform, θ, is then measured using the first IMU which becomes the sensor feedback signal used in the platform controller.

The signal is also then passed into the filter block. This removes the high frequency tremor from signal and leaves the low frequency component to pass through. This becomes the setpoint for the second control system, which is used to move the spoon by means of another, identical motor. Once again, the angle of the spoon, ϕ, is measured using an IMU, and the signal is passed back into the controller thus bringing the spoon under closed loop control. As the movement of the platform effects the spoon and vice versa, the system can be said to be cross coupled. The coupling is represented above as two arrows connecting both mechanics blocks. This effect poses some problems to the control systems that are designed in this project as it can mean that the transfer functions used are less accurate than expected. The block diagram also shows how these blocks are implemented in the testing system. The setpoint, filter and two controllers were created in software and were uploaded to the Arduino Uno microcontroller, the mechanics of the spoon and platform system were dictated by the physical apparatus and the angle measurements were generated using IMUs.

4.2. System Modelling

4.2.1. Derivation of System Equations

As the physical system was now fully realised, the mechanics of the system could be modelled and then used in simulations. Two equations representing torque seen by the hand (platform) and spoon were derived using Newtonian mechanics and the physics behind electronic DC motors. Knowing that high quality motors are being used in the system it can be assumed that the mechanisms are frictionless. The first of these equations was that representing the torque of the platform. To derive this it is first assumed that there are no interactions between the platform and the spoon, that is, there is no friction and the spoon motor is open circuit, then the torque applied to the platform is:

$$J_h\ddot{\theta} = NK_mI_h$$

Here J_h is the moment of inertia of the platform and the $\ddot{\theta}$ is angular acceleration. According to the rotational version of Newton's second law, torque is the product of these two values. The RHS side of the equation is equal to the torque generated by the motor, due to the current flowing in the armature. Here N, K_m and I_h are the motor gearing ratio, the motor torque constant and the armature current respectively. This equation can be put in terms of voltage by replacing I_h using ohms law.

$$J_h\ddot{\theta} = \frac{NK_m}{R}\left(u_h - NK_e\dot{\theta}\right)$$

Here u_h is the voltage applied in the form of a PWM signal, $NK_e\dot{\theta}$ is the back EMF generated from spinning the motor, and R is the terminal resistance of the motor. K_e is called the back-EMF constant and is given in Vrpm^{-1}. Expressing this in terms of rads^{-1}, it is seen that this value equates to the torque constant. For this reason, this value will be replaced by K_m for the remainder of the report. The effects of the spoon motor torque are then considered in the equation. This is found to be a reactive torque that acts on both the spoon and platform in equal and opposite directions, and is given as:

$$T_r = \frac{NK_m}{R}\left(u_s - NK_m(\dot{\varphi} - \dot{\theta})\right)$$

In this case the angular velocity is the difference between $\dot{\theta}$ and $\dot{\varphi}$ as it is the angular velocity with respect to the platform on which the motor is attached, not the horizontal, which is how the angle is measured. This reactive torque acts against the motion of the platform to give:

$$J_h\ddot{\theta} = \frac{NK_m}{R}(u_h - NK_m\dot{\theta}) - T_r = \frac{NK_m}{R}(u_h - NK_m\dot{\theta}) - \frac{NK_m}{R}\left(u_s - NK_m(\dot{\varphi} - \dot{\theta})\right)$$

Rearranging this equation, gives:

$$J_h\ddot{\theta} = \left(\frac{NK_m}{R}\right)(u_h - u_S) - 2\frac{(NK_m)^2}{R}\dot{\theta} + \frac{(NK_m)^2}{R}\dot{\varphi}$$

Finally, the effects of the instability due to the mass of the spoon motor and platform can be added. This torque is simply the force applied multiplied by the radius, where the force being applied is that due to gravity ($M \times g$) and the radius is the distance, d, from the centre of gravity of the platform to the axle about which it spins, multiplied by the sine of θ. This is then also added to the equation and the final torque seen by the platform is:

$$J_h\ddot{\theta} = \left(\frac{NK_m}{R}\right)(u_h - u_S) - 2\frac{(NK_m)^2}{R}\dot{\theta} + \frac{(NK_m)^2}{R}\dot{\varphi} - Mgdsin\theta \qquad \{1\}$$

The torque acting on the spoon is derived in a similar manner. This time the force generated by the motor is the reactive torque calculated above and is now acting with the direction of motion. Once again considering the torque generated due to the current within the armature, the spoon torque is found to be:

$$J_s\ddot{\varphi} = \left(\frac{NK_m}{R}\right)u_S + \frac{(NK_m)^2}{R}(\dot{\theta} - \dot{\varphi}) - mglcos\varphi \qquad \{2\}$$

The added term here is due to the torque created by the mass on the spoon, i.e. the food being held. m, g and l are the mass of the food, the acceleration due to gravity and the distance from the axle to the COG of the mass respectively. Equations {1} and {2} formed the bases of modelling the system mechanics. These were both used in the Simulink models that were created for carrying out simulations and were used in deriving the transfer functions of the system.

4.2.2. Mechanical System Constants

Most of the constants used in the equations above were found by either using the motor data sheet for the 22N28 216E.286 Portescap DC motor or by measuring the dimensions of the physical test system. The more difficult constants to achieve were the moments of inertia of the two bodies and mass and COG position of the platform. To obtain these a 3D model of the system was generated using Solidworks CAD software. With this, an accurate model could be made in which different materials could be used. Details on material were gathered from the mechanical engineering workshop in engineering department of UCC and approximations were made using known masses and dimensions for unknown parts. This 3D model is shown below in figure 4.2. The position of the COG, and moment of inertia about this point could be read using the software. The parallel axis theorem was then used to find the moment of inertia about the platform axle. The mass of the system could also be taken as well as the moment of inertia of the spoon.

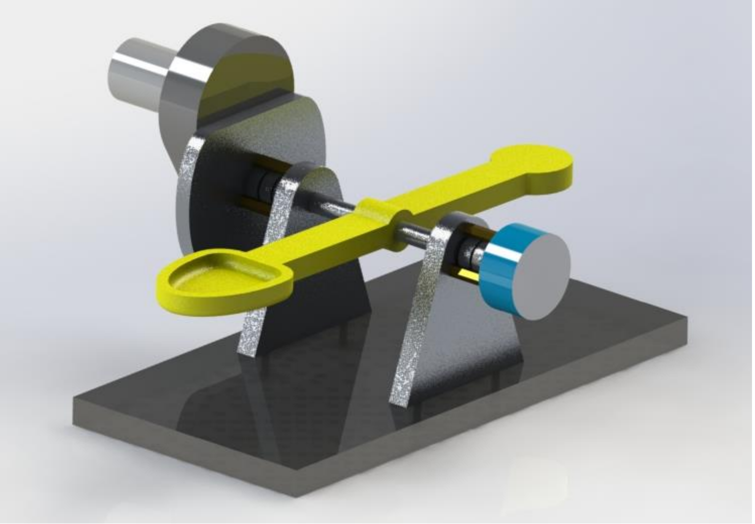


Figure 4.2 – Solidworks Render of platform System

The table below summarises each of the constants gathered for use in the system modelling.

Constant	Symbol	Value	Unit
Platform MOI	J_h	0.00202	kgm^2
Spoon MOI	J_s	0.0005	kgm^2
Motor Torque Constant	K_m	10.2	mNm/A
Gearbox Ratio	N	40:1	—
Terminal Resistance	R	5.8	Ω
Mass of Platform System	M	0.565	kg
Mass of point mass on spoon	m	0.01	kg
Acceleration due to gravity	g	9.81	ms^{-1}
Distance to COG of platform	d	0.0318	m
Distance to point mass on spoon	l	0.1	m

Table 4.1 – System Constants

4.2.3. Simulink Model

With the system mechanics equations formed, a Simulink diagram could be created that represented the entire system. Simulink is an extension of the MATLAB numerical computing environment. The application lets users design multi-domain block diagrams that can represent system mechanics, controller algorithms and mathematical operation. It offers an ideal environment to simulate a system like this as it gives a wide range of tools to aid design and can integrate perfectly with MATLAB by reading and writing data to and from and the command window. The process of building this Simulink diagram began with taking the Laplace transform of the system equations. Looking at the equation representing the platform torque the Laplace transform is:

$$L^{-1}\{J_h\ddot{\theta}\} = L^{-1}\left\{\left(\frac{NK_m}{R}\right)(u_h - u_S) - 2\frac{(NK_m)^2}{R}\dot{\theta} + \frac{(NK_m)^2}{R}\dot{\varphi} - Mgdsin\theta\right\}$$

$$J_h\theta s^2 = \left(\frac{NK_m}{R}\right)(u_h - u_S) - 2\frac{(NK_m)^2}{R}\theta s + \frac{(NK_m)^2}{R}\varphi s - Mgdsin\theta$$

Rearranging this:

$$\theta = \frac{1}{J_h s^2}\left[u_h\left(\frac{NK_m}{R}\right) - u_S\left(\frac{NK_m}{R}\right) - 2\frac{(NK_m)^2}{R}\theta s + \frac{(NK_m)^2}{R}\varphi s - Mgdsin\theta\right]$$

Applying the same process to equation {2} yields:

$$\varphi = \frac{1}{J_s s^2}\left[\left(\frac{NK_m}{R}\right)u_S - \frac{(NK_m)^2}{R}\varphi s + \frac{(NK_m)^2}{R}\theta s - mglcos\varphi\right]$$

These equations were used to make up the Simulink diagram that is shown in Appendix 1 of this report. This purely represents the physical properties of the system and does not include any control systems. The diagram shows how the system is separated into two sections, representing the platform and spoon respectively. In each case there is input of the voltage applied to the DC motor and output of the angle with respect to the horizontal. A summing block can be seen in both sections. These represent the addition that takes place within the square brackets of the two equations derived above. Each value is then divided by the moment of inertia and is integrated twice to achieve the respective angle. As indicated in the diagram, there are several connections between the two systems. These represent the cross-coupling effects that are present due to the reactive torques seen within the motor driving the spoon. This diagram could then be used to design the controller algorithms for each system. The control system was then added into this diagram and simulations were carried out.

4.3. Filter Design

The filter design process proved to be an important part of this project. The performance of this device was highly dependent on correct generation of the setpoint signal for the spoon control system. If the signal coming from the platform IMU was not correctly filtered, then some of the motion of the platform would be seen in the spoon. It was imperative then that the platform angle signal was correctly attenuated. It should be considered as well that the use of a filter will also lead to some degree of phase shift in the signal. The filter implemented should minimise any phase shift as this would lead to delayed and inaccurate tracking of the intentional movement of the user.

4.3.1. Notch Filter

The notch filter, or band-stop filter, was the initial filter design that was researched. The notch filter operates by removing a single frequency element and allowing all other frequency components to pass [30]. The general form for a 2nd order notch filter is as follows:

$$F(s) = \frac{s^2 + {\omega_z}^2}{s^2 + \frac{\omega_p}{Q}s + {\omega_p}^2}$$

Where ω_z, ω_p and Q are the zero frequency, pole frequency and Q factor respectively. The ratio of zero to pole frequency dictates if the filter is a low-pass notch or a high-pass notch. In this case, the low frequencies must not be affected so the two values are set to be equal to each other. In any filter type the Q factor dictates the severity of the roll-off in the stop band. In this case it will then dictate the width of the band being rejected. The bode plot here shows the magnitude and phase characteristics of this filter when different frequency components are

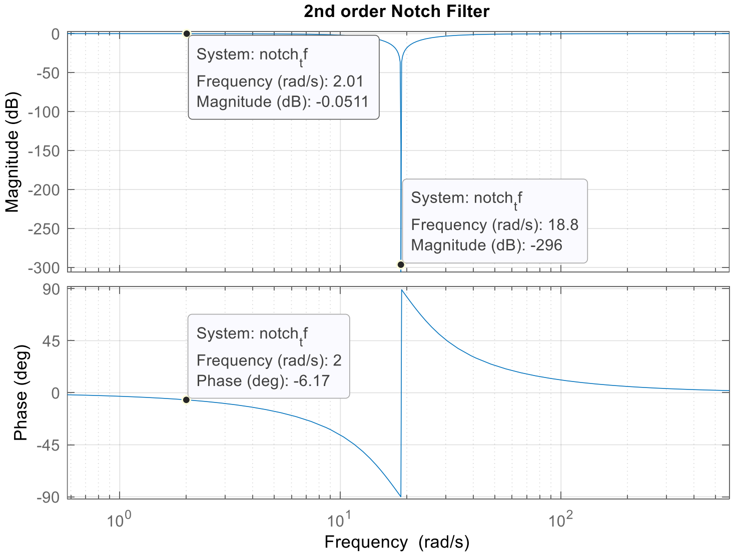


Figure 4.3 – 2nd Order Notch Filter Bode Plot

passed through. The Q factor was set to 1 and the pole and zero frequencies were both set to 18.8 rads^{-1}.

Clearly, this filter has fulfilled the task set out in removing the tremor frequency of 18.8 rads^{-1} (3 Hz), while allowing the low frequency signal to pass through. As well as this, there is minimal phase shift at low frequencies. Implementing this filter should give us close to perfect performance in removing the tremor signal in the system resulting in a clean, low frequency setpoint signal for the spoon controller.

While the notch filter gives ideal behaviour, it does not account for varying hand tremors in the users of this device. For this reason, an alternative filter must be designed that will remove frequencies of a much wider range. A low pass filter was the obvious answer here as it would allow low frequency signals to pass undisturbed but remove high frequencies above a certain cut-off point. While it would be ideal to design a filter that can effectively attenuate the unwanted high frequency component and leave no phase shift at low frequencies, this was not fully possible. As the difference in frequency between the tremor movement and intentional movement was relatively small, higher order versions of low pass filters were used to ensure that the high frequency component was removed correctly (adding poles to system increases the roll-off rate of the magnitude response). These usually introduced a certain amount of phase shift as well. A compromise had to be made here between an allowable degree of motion at high frequencies and phase shift at low frequencies. Considering these limitations, design requirements were estimated using a number of calculations and assumptions.

The allowable amplitude of high frequency components was calculated by considering the force acting on the point mass that the spoon is holding i.e. the food on the spoon. This was achieved by looking at the sinusoidal displacement seen at this point and then finding the maximum acceleration it encounters. It was decided that the tremor signal would oscillate with an amplitude no greater than 0.3 rad. This meant the maximum displacement at a distance of 10 cm from the axis is calculated to be:

$$d = 0.1\sin(0.3) = 0.029\ m$$

Assuming sinusoidal motion of the point mass, the displacement over time is:

$$d(t) = 0.029\sin(\omega t)$$

Where ω is the frequency of oscillation in rads^{-1}. Looking at the 2nd derivative of this function gives the acceleration seen by the point mass over time:

$$a(t) = \ddot{d}(t) = -0.029\omega^2 \sin(\omega t)$$

If the spoon is to oscillate at the tremor frequency of approximately 18 rads^{-1} then the maximum force applied to the point mass is calculated as follows:

$$F = ma = -0.029m\omega^2 \sin(\omega t)$$

This gives a maximum value when: $\sin(\omega t) = -1$

$$\therefore F = -0.029m\omega^2(-1)$$

$$F = 0.029(18^2)m = 9.4m \text{ N}$$

Where m is the mass being held by the spoon. As seen here, under the current conditions, the force applied to this mass due to the movement of the spoon is close to that of the of force due to gravity (9.81m N). With this it is clear that the mass on the spoon will spill due to the oscillation of the spoon. It was decided that a force of about one twentieth of this would be enough to prevent any spillages. This meant a displacement, due to tremor, of approximately 1.45 mm was allowable at the end of the spoon. With this, the angle could be calculated and finally, the level of attenuation required by the filter.

$$\varphi = sin^{-1}\left(\frac{0.145}{10}\right) = 0.0145 \text{ rad}$$

$$A = 20log_{10}\left(\frac{0.0145}{0.3}\right) \approx -26 \text{ dB}$$

A phase shift of approximately of 35° was decided upon as it gave a relatively small delay if it is assume that the intentional movement has a frequency of 2 rads^{-1}. A delay of approximately 0.3 seconds would be seen here which was calculated as follows:

Period of oscillation:

$$T = \frac{1}{f} = \frac{2\pi}{\omega} = \frac{2\pi}{2} = \pi \approx 3.14 \text{ s}$$

$$t_D = 3.14 \times \frac{35}{360} = 0.305\, s$$

Although it would be best to design filter with no phase shift, it was decided that a time delay of 0.3 s was permissible as it would likely still allow the spoon to function correctly and remain helpful for the user.

4.3.2. Butterworth Filter

The first low pass filter that was considered was in the form of a Butterworth filter. Butterworth filters are an example all pole filters, i.e. they have no s term in the numerator of the transfer function leaving only poles in the denominator. The filter is designed with goal of having minimal loss at low frequencies and is, therefore, often referred to as a maximally flat magnitude filter [30]. It designed by looking at the general form of magnitude squared function for an nth order all-pole low pass filter:

$$G(\omega^2) = \frac{1}{c_0 + c_1\omega^2 + c_2\omega^4 + \cdots + c_n\omega^{2n}}$$

In order to achieve unity gain at low frequencies, the denominator of this equation is kept as close to 1 as possible for small values of ω. This is achieved by letting c_0 = 1 and then setting each derivative of the denominator to 0 at ω = 0. This then forms the magnitude squared function for the nth order Butterworth filter.

$$G(\omega^2) = \frac{1}{1 + \omega^{2n}}$$

With this equation the transfer function can then be found. For the 2nd order low pass filter this becomes:

$$G(\omega^2) = \frac{1}{1 + \omega^4}$$

Giving the transfer function:

$$F(s) = \frac{1}{1 + \sqrt{2}s + s^2}$$

This is identical in form to a normalised 2nd order low pass filter with the Q factor set to $1/\sqrt{2}$. This is suitable for our device as it means there is no gain seen at the corner frequency which might hinder the tracking of intentional movement. The filter can then be frequency scaled to give a desired corner frequency by replacing ω with ω/ω_c.

$$F(s) = \frac{1}{1 + \sqrt{2}\frac{s}{\omega_c} + \left(\frac{s}{\omega_c}\right)^2}$$

$$F(s) = \frac{{\omega_c}^2}{{\omega_c}^2 + \sqrt{2}\omega_c s + s^2}$$

The bode plot shown below shows the magnitude and phase characteristics of this filter as the frequency of the signal is increased.

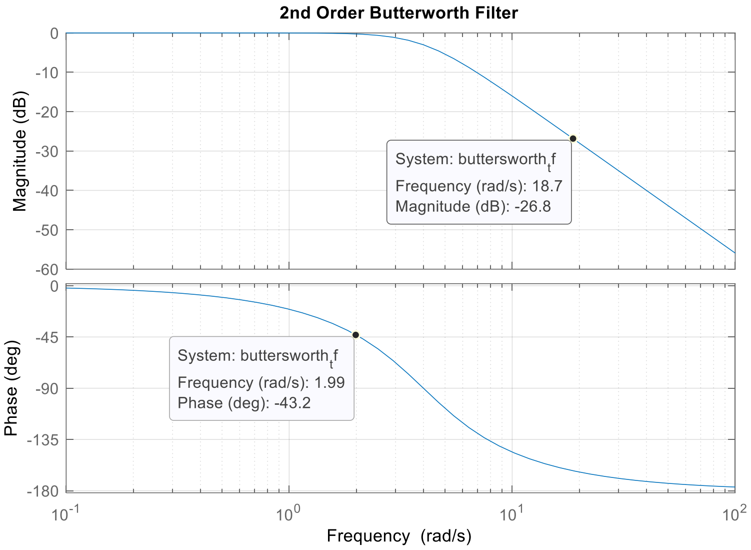


Figure 4.4 – 2nd Order Butterworth Filter Bode Plot

While the attenuation seen here is acceptable, the phase shift could be improved greatly. For this reason, a higher order alternative was designed. Knowing the design parameters, outlined earlier, the corner frequency and order number could be calculated. The design parameters are as follows:

	Gain, A (dB)	Frequency, ω (rads^{-1})
Pass band	-0.1	2
Stop band	-26	18.8

Table 4.2 – Filter Design Parameters

With these, we can fill into the following equation to find the order of the filter required:

$$n = \frac{log_{10}\left(\frac{10^{0.1A_{min}}-1}{10^{0.1A_{max}}-1}\right)}{2log_{10}\left(\frac{\omega_s}{\omega_p}\right)} = \frac{log_{10}\left(\frac{10^{0.1(26)}-1}{10^{0.1(0.1)}-1}\right)}{2log_{10}\left(\frac{18.8}{2}\right)} = 2.17 \approx 3$$

With this we can find the denominator of the normalised transfer function by reading from the Butterworth design tables. Frequency scaling is applied again to find the transfer function:

$$F(s) = \frac{{\omega_c}^3}{({\omega_c}^2 + \omega_c s + s^2)(\omega_c + s)}$$

The required corner frequency is calculated to be:

$$\omega_c = \omega_p 10^{-\left(\frac{log_{10}(10^{0.1A_{max}}-1)}{2n}\right)} = 2 \times 10^{-\left(\frac{log_{10}(10^{0.1(0.1)}-1)}{2(3)}\right)} = 3.742\ \text{rads}^{-1}$$

The bode plot for this filter is shown below:

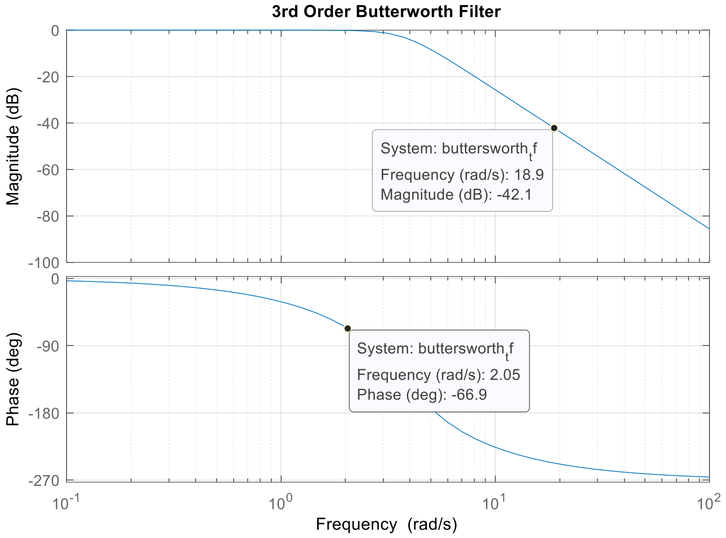


Figure 4.5 – 3rd Order Butterworth Filter Bode Plot ($\omega_c = 3.7\ rads^{-1}$)

Like the previous example, the attenuation at high frequencies is optimal here but the phase shift at low frequencies could still be massively improved. This time however, the corner frequency can be scaled further while maintaining an acceptable degree of attenuation at the tremor frequency.

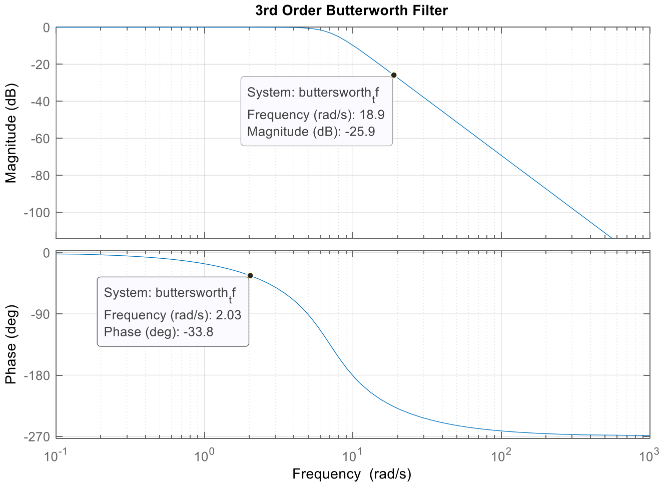


Figure 4.6 – 3rd Order Butterworth Filter Bode Plot ($\omega_c = 7\ rads^{-1}$)

The corner frequency was now increased to 7 rads^{-1}. This time the phase shift and attenuation levels and much more reasonable with values of 33° and -25.9 dB at 2 rads^{-1} and 18.9 rads^{-1} respectively. This filter design was chosen to be implemented and tested in simulations.

4.3.3. Chebyshev Filter

The maximally flat approximation used in the Butterworth filter means that the error in magnitude response is near to prefect at low frequencies but becomes progressively worse as the frequency is increased. The Chebyshev filter aims to remedy this problem by aiming for a minimum amount of error across the specified range. For this reason, an equiripple criterion is applied when determining the normalised transfer functions. The benefit of using a Chebyshev is the high rate of magnitude drop-off in the passband. This is at the expense of ripple in the passband and a less linear phase characteristic. As a certain level of attenuation would be acceptable in the pass band it was decided that the Chebyshev filter would be investigated. Increasing the allowable ripple in the pass band of this filter type will improve the magnitude response after the corner frequency, making the slope steeper. Considering this knowledge, a value of 0.5 dB was chosen as a reasonable amount of ripple in the passband that would still allow effective performance. This would only account for about 10% of the amplitude of oscillation.

The design process for the Chebyshev filter was similar to that of the Butterworth. An initial order number of 3 was calculated but it was decided that a higher order filter would be used as it would give a sharper roll-off in the stop band, thus allowing the corner frequency to be pushed closer to the 18.8 rads^{-1}, meaning less of a phase shift seen at low frequencies. The transfer function for a 5th order low pass Chebyshev filter is as follows:

$$F(s) = \frac{0.1789{\omega_c}^5}{(s + 0.3623\omega_c)(s^2 + 0.2239\omega_c s + 1.0358{\omega_c}^2)(s^2 + 0.5862\omega_c s + 0.4768{\omega_c}^2)}$$

A corner frequency of 7 rads^{-1} was chosen and the following bode plot was generated.

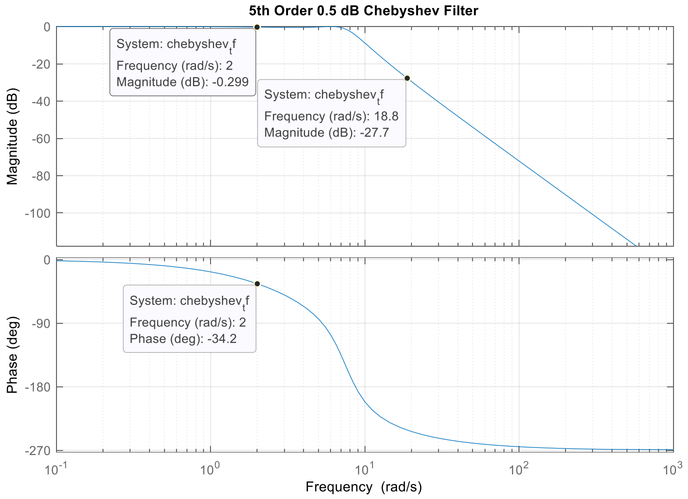


Figure 4.7 – 5th Order Chebyshev Filter

As the graph shows, the phase shift at low frequency and the attenuation at high frequency are within the acceptable limits. Moreover, the magnitude response at 2 rads^{-1} is only -0.299 dB, meaning attenuation of intentional movement should not be a problem. This filter design will then also be tested along with the Notch and Butterworth filters.

4.4. Controller Design

The following section outlines the steps taken to obtain the controller designs for both the platform and spoon systems. Two types of controller were chosen for each. The first of these was the PID control system. Its simplicity and effectiveness meant that would stand as good benchmark in the project. In hope the of improving on the performance of this, a phase lead compensator was also designed for each of the systems.

4.4.1. Platform Controller

The control system was designed by first setting up the feedback loop in the Simulink diagram. This can be seen below in figure 4.8 below. As can be seen here, the controller is processing the error that is produced from the setpoint and feedback signal, creating a signal that is to drive this error to zero. The signals that are coming from the spoon system are now to set to zero as it is assumed that the spoon is motionless while the platform controller is being tuned.

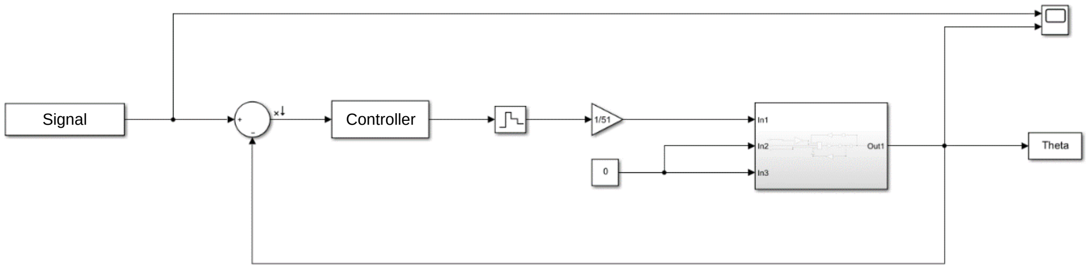


Figure 4.8 – Platform Control System

A zero-order hold has now also been added which ensures the digital signal produced by the controller is held for the duration of the time between samples. As well as this a gain of 1/51 is added. This is because the controller will generate an integer between 0 and 255 to control the PWM signal that will create a voltage between 0 and 5 V.

$$255 \times G = 5\,V \quad \rightarrow \quad G = \frac{5}{255} = \frac{1}{55}$$

A. PID Controller

The PID controller was designed by first using the Ziegler-Nichols tuning method and then making small alterations to the gains of the transfer function to improve the performance further. The transfer function for a PID controller is shown below. Three gains, K_P, K_I and K_D are varied to improve the performance of the system.

$$G_{PID}(s) = K_P + \frac{K_I}{s} + K_D s$$

Ziegler-Nichols tuning is carried out by first setting the values of K_I and K_D to zero and then adjusting the value of K_P until sinusoidal behaviour is seen when an impulse is applied to the system. Looking back on the discussion in chapter 2 on plotting pole and zeroes, this step is equivalent to placing the dominant closed loop poles of the system on the imaginary access, thus producing sinusoidal behaviour. The graph below shows the plot of the platform angle, θ, and the setpoint signal, when this is achieved.

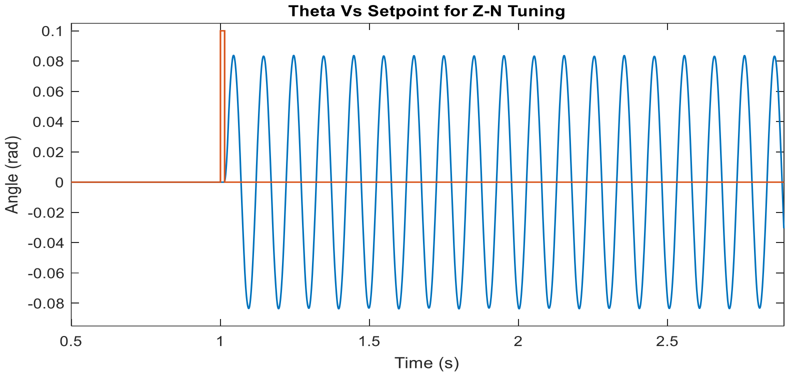


Figure 4.9 – Impulse Response with Ultimate Gain

The gain values were then found using the Ziegler-Nichols table shown below. Here the K_U is ultimate gain, which is the value of K_P required to produce sinusoidal behaviour, and T_U is the period of oscillation.

Control Type	K_P	K_I	K_D
P	$0.5K_U$	—	—
PI	$0.45K_U$	$\frac{0.54K_U}{T_U}$	—
PD	$0.8K_U$	—	$\frac{K_U T_U}{10}$
PID	$0.6K_U$	$\frac{1.2K_U}{T_U}$	$\frac{3K_U T_U}{40}$

Table 4.3 – Zeigler Nichols Tuning Equations

With this the values of K_P, K_I and K_D were calculated to be 3947.4, 78948 and 49.343 respectively. Finally, some small alterations were made to improve the performance of the controller. Figure 4.10 shows the performance observed when a sinusoidal signal with a frequency of 18.8 rads^{-1} was applied at the setpoint.

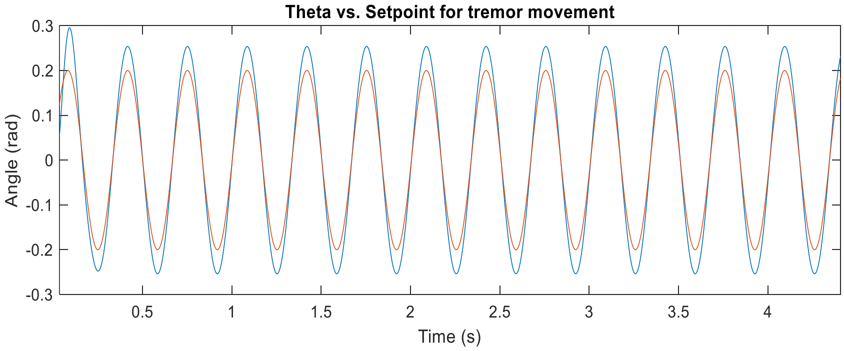


Figure 4.10 – Platform Angle vs. Setpoint for tremor Movement

B. PHASE LEAD COMPENSATOR

Applying the root locus design method, the transfer function of the platform system must be found. By making a few assumptions, a simplified version of the transfer functions was derived. This starts with taking the equation for the platform torque and assuming that there is no reactive torque present due to spoon motor:

$$J_h\ddot{\theta} = \frac{NK_m}{R}\left(u_h - NK_m\dot{\theta}\right) - Mgdsin\theta$$

Assuming that the platform is oscillating with a small amplitude, then $sin\theta$ can be taken as just θ.

$$J_h\ddot{\theta} = \frac{NK_m}{R}u_h - \frac{(NK_m)^2}{R}\dot{\theta} - Mgd\theta$$

Taking the Laplace transform of this and rearranging, the transfer function can be found:

$$J_h\theta s^2 = \frac{NK_m}{R}u_h - \frac{(NK_m)^2}{R}\theta s - Mgd\theta$$

$$\rightarrow \left[J_h s^2 + \frac{(NK_m)^2}{R}s - Mgd\right]\theta = \frac{NK_m}{R}u_h$$

$$\rightarrow \frac{\theta}{u_h} = \frac{\frac{NK_m}{R}}{\left[J_h s^2 + \frac{(NK_m)^2}{R}s - Mgd\right]} \times \frac{1}{51}$$

In the final step, the gain of 1/51 is added in as this is now part of the transfer function as well. As indicated here, a transfer function is now derived that describes the characteristic between the motor voltage and the angle of the platform. Once this was achieved, a digital version of the transfer function could be found using MATLAB. In transforming the system to the digital z-domain, a zero-order hold is effectively being added to the transfer function. The transfer function of the zero-order hold is given as:

$$G_{ZOH}(s) = \frac{1 - e^{-sT_s}}{s}$$

In this equation T_s is the sampling period. This process is simplified by using the continuous to discrete function on MATLAB. Filling in the constants for the system, the digital transfer function is found to be:

$$F(z) = \frac{5.85e^{-5}z + 5.49e^{-5}}{z^2 - 1.84z + 0.8255}$$

A root locus for this function was then plotted, which can be seen below in figure 4.11.

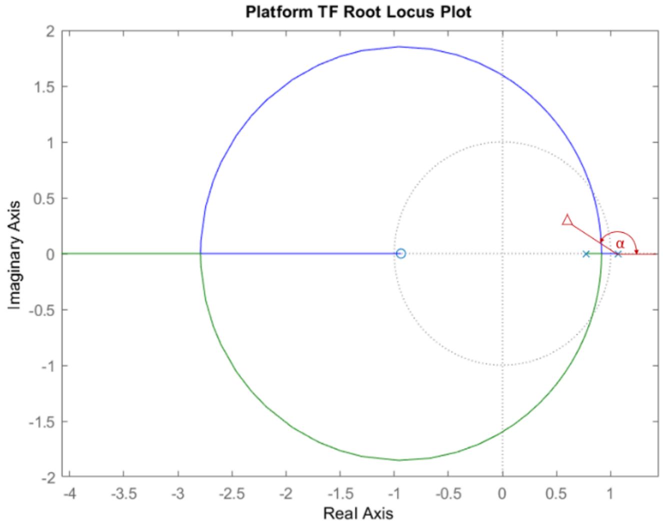


Figure 4.11 – Platform Root Locus

This root locus plot is a bit different to that seen in chapter one. This is because it is now being plotted in the z-domain. Positions on the s-plane are transformed to the z-plane by using the identity: $z = e^{sT}$. The result of this is that plots running along vertical asymptotes to plus and minus infinity now follows a circular path like that shown above. As well as this, the imaginary axis has now become the unit circle shown as a dotted line in the graph. The result of this is that this is now the margin of stability, with closed loop poles within the circle showing a stable system. Furthermore, the closer the closed loop poles are to the origin, the more stable the system is.

Knowing this, the addition of a phase lead compensator will bring the root locus plot closer to the origin and a stable system can then be found. If the position of the desired closed loop pole is known, then the positions of the required pole and zero of the phase lead compensator can be found using the angle between the real axis and the desired pole location, at each of the pre-existing pole and zero positions. An example of this is shown in figure 4.11. Here, the desired pole location is shown as a triangle and the angle at the pole is α. In the case of this project, a damping factor of 0.5 was desired, which gave a closed loop pole location of the $0.65 + 0.375j$. The angle between the known pole and zero locations could then be calculated using the arctangent function. As well as this the position of the compensator zero was chosen to be directly below this point on the real axis as it would give the greatest attractive benefit at this location. With this, the following identity could be used to find the position of the compensator pole.

$$\sum \alpha - \sum \beta = 180$$

Where α is a pole angle and β is a zero angle. By filling into this, the compensator pole angle, α_p, was found.

$$\alpha_1 + \alpha_2 + \alpha_p - \beta_1 - \beta_2 = 180$$

$$108.43 + 137.55 + \alpha_p - 13.28 - 90 = 180 \quad \rightarrow \quad \alpha_p = 37.302^{\mathrm{o}}$$

Using the tangent function, the position of the pole was then found to be at: $z = 0.157$.

Using this phase lead compensator, with pole at 0.65 and zero and 0.157, a new root locus diagram could be plotted, as shown in figure 4.12 below. The root locus is now observed to go through the desired closed loop pole location. Finally, the gain of the phase lead compensator was found, using Sisotool, another application provided by MATLAB that gives the ability to view and edit transfer function plots. The gain was found to be 2500, which could be edited in the Simulink model to achieve better results if needed.

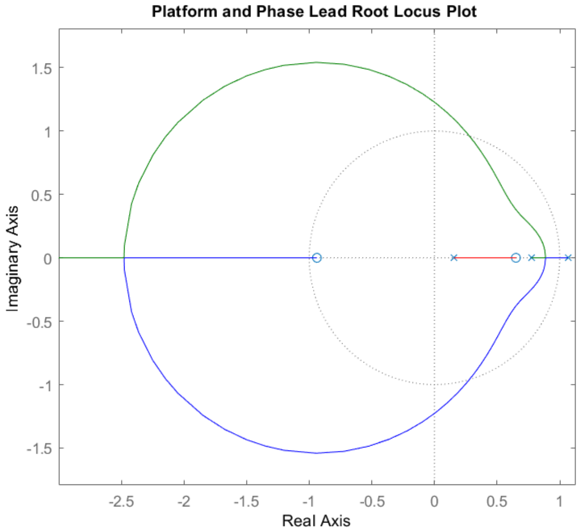


Figure 4.12 – Platform Root Locus with Lead Compensator

4.4.1. SPOON CONTROLLER

The controller design process used for the spoon system was much the same as the platform. It began with establishing the feedback system required for closed loop control, a diagram of which can be seen in the figure 4.13 below. This time, the platform movement was set to zero so any cross-coupling connections that were present between the two systems were set to zero as well. Once again, a zero-order hold, and a gain block were added to mimic the effects of the digital micro-controller.

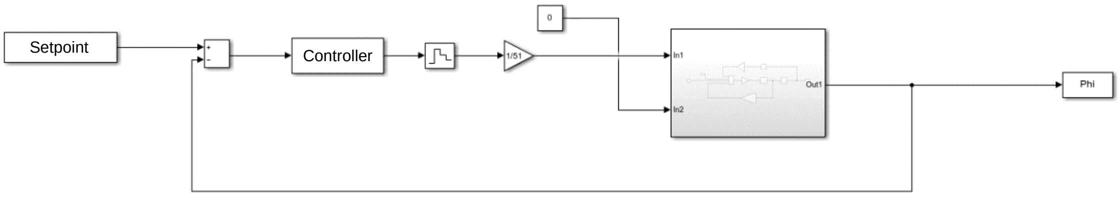


Figure 4.13 – Spoon Control System

A. PID CONTROLLER

The Zeigler Nichols Method was used again here for tuning the PID controller. It was decided that PI and PD controllers would also be designed and simulated as they might perform better when simulated and could mean less issues in implementation on the Arduino controller. The method of tuning remained the same in that an impulse response was applied to the system and the proportional gain K_P was altered until sinusoidal behaviour was observed. This was found to be at a value of 3536 giving a period of oscillation of 0.072 seconds. The Ziegler Nichols tuning identities shown in table 4… were used to find the gains required for each controller. These gain values are summarised in the table below.

Control Type	K_P	K_I	K_D
P	1768	—	—
PI	1591.2	26520	—
PD	2828.8	—	25.459
PID	2121.6	58933.3	19.09

Table 4.4 – Gain Values for P, PI, PD and PID Controllers

To ensure these values were correct a simulation was run by applying a low frequency sinusoidal signal as the setpoint, representing the intentional movement the spoon would have track. The setpoint and output signals are shown below showing ideal performance using this controller.

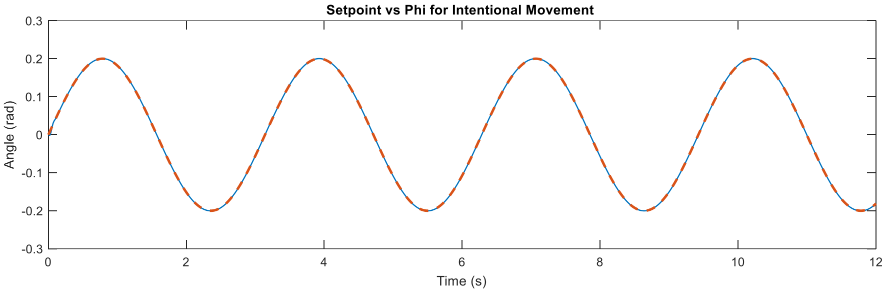


Figure 4.14 – Spoon Angle vs Setpoint for non-tremor movement

B. PHASE LEAD COMPENSATOR

A phase lead compensator was once again designed using the root locus method, as outlined in the previous section of this report. To achieve this, the transfer function of the spoon system had to first be formed. Equation {2} was used this time which represented the torque seen by the spoon. The angular velocity of the platform spoon could now be taken to be zero giving the following equation:

$$J_s\ddot{\varphi} = \left(\frac{NK_m}{R}\right)u_S - \frac{(NK_m)^2}{R}\dot{\varphi} - mglcos\varphi$$

As the mass, m, placed on the spoon is small, it was assumed that the torque due to this was zero. The Laplace transform could then be taken, and the equation could be rearranged.

$$J_s\varphi s^2 = \frac{NK_m}{R}u_s - \frac{(NK_m)^2}{R}\varphi s$$

$$\rightarrow \quad \left[J_s s^2 + \frac{(NK_m)^2}{R}s\right]\varphi = \frac{NK_m}{R}u_s$$

$$\rightarrow \quad \frac{\varphi}{u_s} = \frac{\frac{NK_m}{R}}{\left[J_s s^2 + \frac{(NK_m)^2}{R}s\right]} \times \frac{1}{51}$$

The transfer function is again multiplied by the gain of 1/51, representing the conversion from an 8-bit integer value to a voltage between 0 and 5 V. The transfer function was then converted to a discrete function using the continuous-to-discrete (C2D) command in MATLAB. As before, the root locus plot was generated for the system which was to be altered with the addition of the phase lead compensator. The desired closed loop pole location was then chosen, and the pole and zero values of the phase lead compensator were calculated using this desired pole location. A damping factor of 0.3 was required for this controller which yielded a desired pole location at $0.35 + 0.65j$. Using the angles at each pole and zero, the position of the compensator pole could be calculated.

$$\alpha_1 + \alpha_2 + \alpha_p - \beta_1 - \beta_2 = 180$$

$$99.7 + 135 + \alpha_p - 30.06 - 90 = 180 \quad \rightarrow \quad \alpha_p = 65.36^{\text{o}}$$

This meant the pole location of the compensator would need to be placed at 0.052. The gain of the controller was found to be 2600, giving the compensator transfer function:

$$C(s) = 2600\frac{z - 0.35}{z - 0.052}$$

The addition of this compensator gave the following root locus plot.

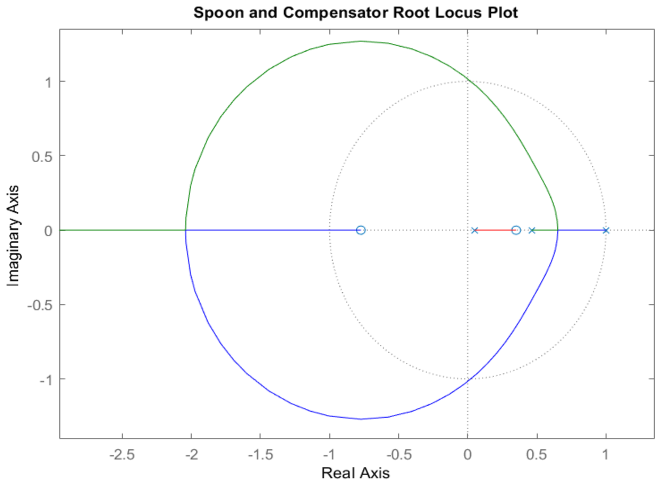


Figure 4.15 – Spoon Root Locus with Lead Compensator

4.5. Arduino Implementation

4.5.1. Sampling Time

While, in theory, the Arduino can sample analog inputs at a rate of 1.23 MHz (crystal frequency is 16 MHz and 13 cycles are required to sample), the sampling rate of this system was limited by the serial communication used in recording data. While the final device does not require communication to a serial monitor, the process of designing it does. This issue was most evident when calibrating the two IMU sensors. In this case three strings had to be printed to the serial monitor during each cycle of the code; the analog input measuring the potentiometer voltage and the IMU readings for the accelerometer and gyroscope. Although a baud rate of 9600 bps was used, each character required 10 bits to be sent to the serial monitor, so the length of time it took to send 1 character was 1.04 ms. As multiple characters were required for each string, the cycling time of the Arduino code became limited by this. A possible solution that was investigated was to store the data in an array and then print the values to the monitor after the test was finished. This did not work, however, as the SRAM on the Arduino Uno was not large enough. In the end, a reasonable sampling time of 13.5 ms was chosen which had to be used for the rest of the experiments as the IMU was calibrated for this value. This was achieved by adding delays to increase the cycling time.

4.5.2. Digitization of Control system

In order to implement each of the controllers and filters designed, they first had be to digitized, that is represented in terms of samples, k. This was achieved using the matched pole zero method. To show an example of this, the notch filter, designed in section 4.3.1, was digitized as follows. Once the transfer function was established it could be transformed to the z-domain using the continuous to discrete function in MATALB, with a sample time set to 13.5 ms:

$$F(z) = C2D\left(\frac{s^2 + 18.8^2}{s^2 + 18.8s + 18.8^2}\right) = \frac{z^2 - 1.941z + 0.9976}{z^2 - 1.719\text{z} + 0.7758} = \frac{R_s}{\theta}$$

Knowing that this is the transfer function between the platform angle, θ, and the setpoint signal for the spoon, R_s, then the equation can be rearranged to find the setpoint in terms of the angle.

$$\theta(z^2 - 1.941z + 0.9976) = R_s(z^2 - 1.719\,\text{z} + 0.7758)$$

$$\theta z^2 - 1.941\theta z + 0.9976\theta = R_s z^2 - 1.719R_s\text{z} + 0.7758R_s$$

Dividing across by z^2 shows how this equation is represented as delays.

$$\theta - 1.941\theta z^{-1} + 0.9976\theta z^{-2} = R_s - 1.719R_s z^{-1} + 0.7758R_s z^{-2}$$

Finally, this can then be represented in terms of signal samples, k, where k is the current sample and $k-1$ is the last sample, etc.

$$\theta(k) - 1.941\theta(k-1) + 0.9976\theta(k-2) = R_s(k) - 1.719R_s(k-1) + \ 0.7758R_s(k-2)$$

$$\rightarrow \quad R_s(k) = \theta(k) - 1.941\theta(k-1) + 0.9976\theta(k-2) + 1.719R_s(k-1) - \ 0.7758R_s(k-2)$$

This now represents the current setpoint signal that is the output of the filter in terms of previous values of θ and R_s. This could then be coded in the Arduino IDE is a straightforward manner.

This process could also have been carried out for the PID transfer functions, but in order to allow efficient tuning of the algorithms when implemented on the Arduino, it was decided that digital integration and derivation functions would be used along with the gains that could easily be altered from the IDE. Digital integration was carried by using Euler's forward difference method, which was discussed in detail in chapter 2 of this report. Digital differentiation then is just a matter of finding the slope of the error signal. This was achieved by taking the difference between the current error value and the last error value, and dividing it be the sampling period.

4.5.3. General Outline of Arduino Code

Appendix 2 shows the final code used on Arduino Uno that implemented the entire control system. The following table briefly outlines what each section is doing.

Lines	Code Description
1 - 5	Include libraries to be used in the code.
10 - 16	Assign names to IMUs and define the registers they are using.
18 - 26	Assign Motor Control calls to required digital out pins.
29 - 74	Define variables and constants, and assign initial values.
82 - 102	Initialise IMUs and ensure communication is correct. Print to monitor if not.
104 - 108	Motor analog output setup.
119 - 159	First Stage of test run. Initial value of each of the angles is calculated.
162 - 187	Start stage 2 of test run. Begin reading gyroscope and accelerometer values from each of the IMUs.
191-199	Calculate angles, using gyroscope with Euler's forward difference method of integration, and arctangent of accelerometer readings. Use estimator to compile the two values.
204	Define setpoint of platform using two sinusoid signals.
207	Define setpoint of spoon using theta values passed through notch filter.
210 - 213	Assign values to the variables representing previous samples taken.
215 - 219	Calculate errors and apply digital integration and derivation for PID control.
222 - 223	Store error values for use in the later integration and derivation calculation.
227 - 228	Find correction signals to be applied to motor using controller algorithms.
231 - 257	Find magnitude of correction factor and cap values at 255
261 -283	Apply correction magnitude and direction to each of the motors, and add delay.

Table 4.5 – Arduino Code Description

5. Results

5.1. Simulation Results

The finalised Simulink can be found in Appendix 3 of this report. Here the entire system has been modelled with both control systems fully implemented along with the filter. It can be seen here that the platform angle is now being fed back into the control system as well as the filter, where the tremor frequency signal of 18.8 rads^{-1} is being removed and the low frequency intentional movement is passing which then becomes the setpoint for the second control system. A saturation block has now also been added after each of the controllers. This is included to prevent the output of the controller from going beyond ± 255 which is what the Arduino PWM signal is capped at. This means that 5 V is the maximum voltage that can be applied to the motors in the simulation, thus representing the real-world system.

The setpoint signal for each of the tests carried out was made up of two frequency components, a 18.8 rads^{-1} signal that represented the tremor behaviour and a 2 rads^{-1} signal that represented the intentional movement of the user. The performance of the system was dictated by how much of the tremor movement was removed from the output.

5.1.1. Platform Control

To ensure that the repeatable tests could be carried out when testing and showing the performance of the system, the platform controller had to perform reasonably well. The following sections show how the PID and phase lead compensator perform in the simulations. In each case the graph shown is a plot of the setpoint signal and the platform angle, θ.

A. PID Control

The following graph shows the observed behaviour of the platform when the PID design in the previous chapter was used.

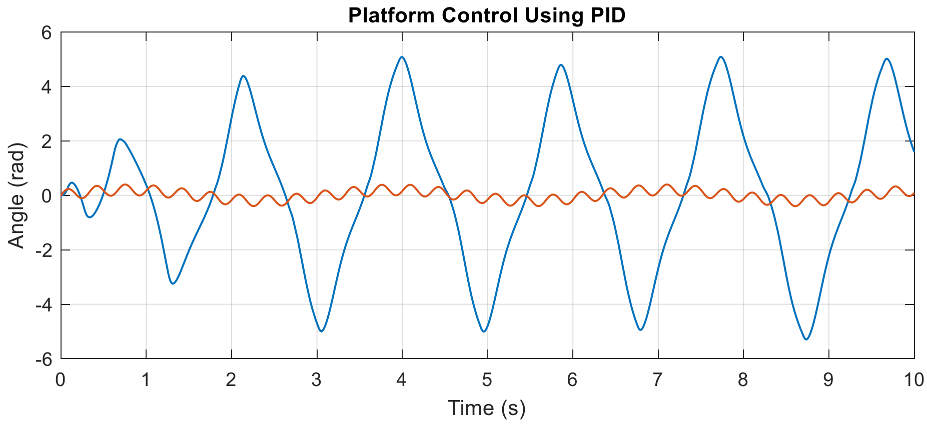


Figure 5.1 – Initial Platform PID Performance

The performance seen here was unexpected as the platform began to oscillate with a huge amplitude. This would equate to the platform spinning around its axle numerous times before spinning in the opposite direction. The reason for this seemed be due to the saturation that is now preventing the controller from outputting a voltage above 5 V. This could be leading to integral windup which is then causing the system to overshoot massively which in turn causes the platform angle to go beyond 90° at which point the top heavy nature of the platform will cause it the spin further, thus adding to problem. This spinning continues to occur until the integral becomes large enough to overcome this. To account for this, smaller gains were applied to the P and I functions and a saturation block was also added to prevent the device from spinning more that 15° in either direction. Appendix 4 shows the performance observed when K_P was set to 2,000 and K_I was set to 20,000. Here the system was overshooting at each of the crests and troughs. In an attempt to remedy this, the derivative gain was increased to 150 at which point the following performance was observed:

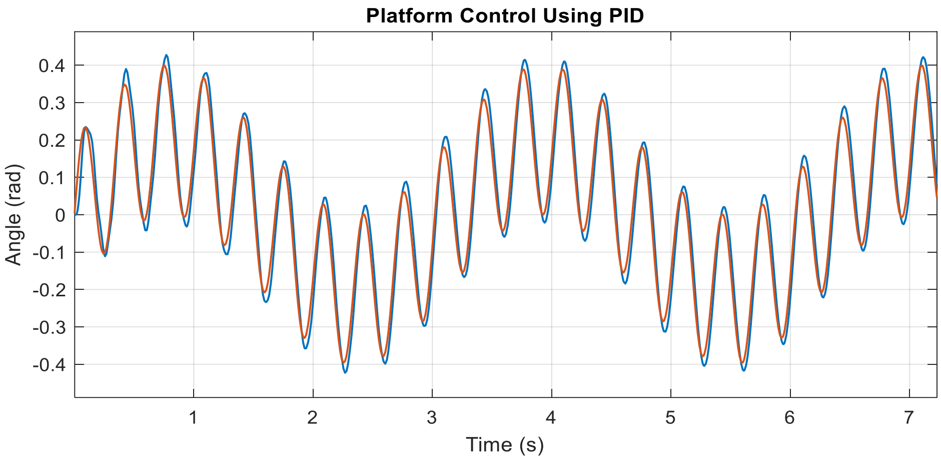


Figure 5.2 – Final Platform PID Performance

Now it can be seen that the two plots are nearly identical with minimal overshoot observed.

B. Phase Lead Compensator

The same test was then carried out for the phase lead compensator. The following plot was generated from this simulation after some tuning was carried out on the compensator gain.

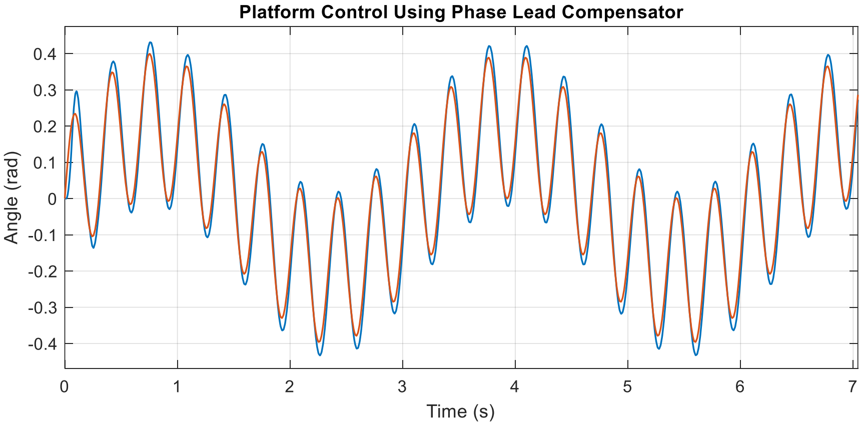


Figure 5.3 – Final Platform Compensator Performance

The compensator used to achieve this plot had the same poles and zeroes as the compensator that was designed. The gain however was increase to a value of 9000 which gave markedly better performance. This shows that the desired close loop pole that was chosen may not have been at the optimal position for control of this system.

5.1.2. Filter Performance

Each of the filters were tested by probing looking at the plot of the platform angle, θ, and the setpoint signal for the spoon control system. With this the performance of each type could be analysed.

A. Notch Filter

The performance of the notch filter is shown in figure 5.4. As expected, the system gives near perfect response with a minimal amount phase shift seen at the low frequency and complete attenuation of the tremor signal. This would form the benchmark for the other filters and would be the filter used in testing the spoon controller performance.

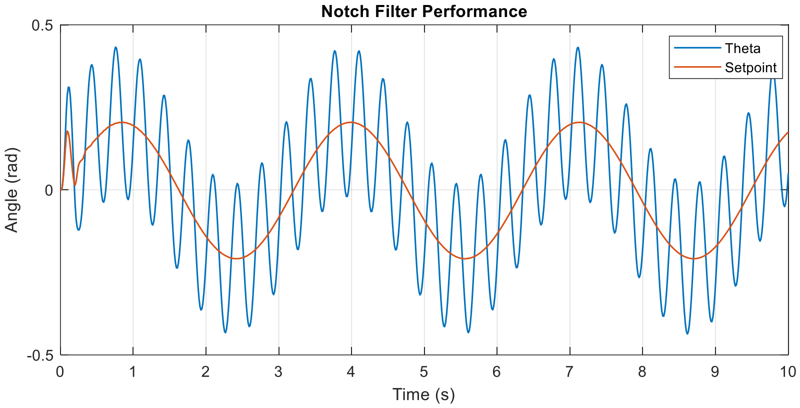


Figure 5.4 – Notch Filter Performance

In order to see the effects of a changing tremor frequency, the high frequency was raised to 25 rads^{-1} (4 Hz), a common tremor frequency seen in both ET and PD. This gave the performance seen below. The reason for choosing a low pass filter instead of a notch filter is evident here as the high frequency component is not attenuated enough.

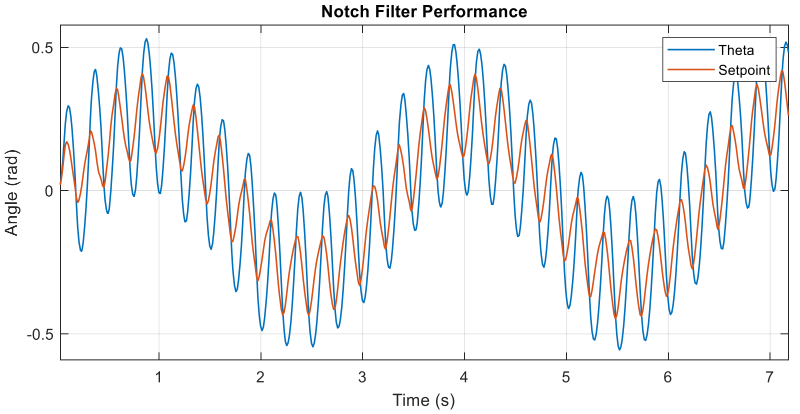


Figure 5.5 – Notch Filter Performance (Tremor of 25 rads^{-1})

B. Butterworth Filter

The 3rd Butterworth filter was then tested, the performance of which is shown below. As expected a phase shift of about 30° is seen and a small ripple is still present at high frequencies. This is relatively small in comparison to the 0.2 rad oscillations seen in the platform angle signal, so this behaviour is acceptable.

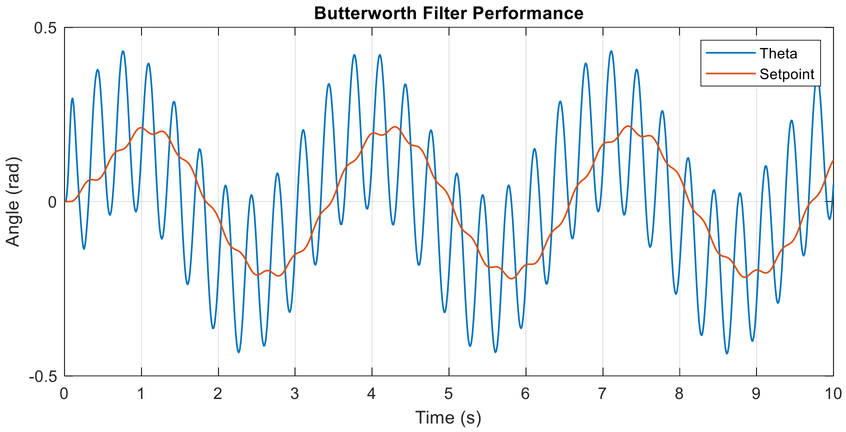


Figure 5.6 – Butterworth Filter Performance

C. Chebyshev Filter

Lastly, the 5th Chebyshev filter performance was observed, as shown in figure 5.7. Once again, a degree of phase shift is seen here of about 30°. A certain amount of the tremor frequency also passes through the filter but to less of an extent than the Butterworth filter.

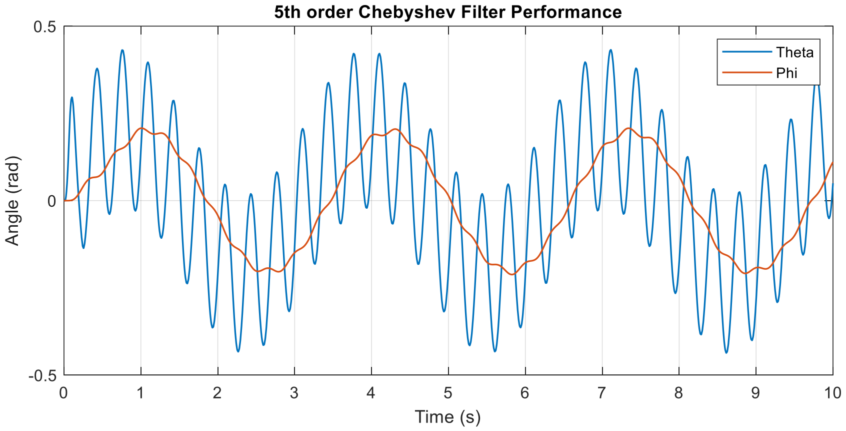


Figure 5.7 – Chebyshev Filter Performance

It is worth noting that when the frequency of the tremor is increased for each of the low pass filters, the level of attenuation at high frequency is improved. The results of these simulations can be found in appendices 5 and 6 of this report. In both of these cases the tremor is increased to 25 rads^{-1} resulting in almost complete attenuation of the high frequency signal. This shows that the low pass filters may be more appropriate for a patient with varying hand tremor.

5.1.3. SPOON CONTROL

The performance of the spoon controller was inspected in each case by comparing the trace of the platform angle, θ, and spoon angle, φ. This was the primary test of the performance of a system as a whole and these would be the direct metrics that would be inspected in a real device. In order to ensure the best results for these tests, the notch filter was used as it gave perfect removal of the tremor frequency. If any irregularity were present here it could be assumed that it wasn't due to the performance of the filter. For each controller, the values calculated in the design section of this report were first used after which the controller was tuned to ensure best performance in signal tracking and transient response.

A. PID CONTROL

The first controller that was tested in the simulations was the PID controller. This would form the baseline for tests regarding the PI, PD and phase lead controllers. The best performance was achieved using the PID is shown below in figure 5.8. Here it is seen that the tremulous motion has not been fully attenuated from the signal, an unexpected result considering the ideal behaviour of the filter. After some investigation it was found that this was due to the reactive force present between the two systems at the spoon motor, and more specifically, the movement of the platform being transferred to the spoon through the motor. Disconnecting the platform to spoon connection on the Simulink gave the performance seen in Appendix 7. The effects of this cross-coupling can be seen clearly with this. In order to measure the attenuation for high frequencies, the low frequency component was disconnected, leaving just the high frequencies, as shown in Appendix 8. Here the attenuation was found to be -8.27 dB, giving about 85% removal of tremor motion. This was reasonable considering the performance of the Liftware and GYENNO spoons which were 70% and 85% respectively.

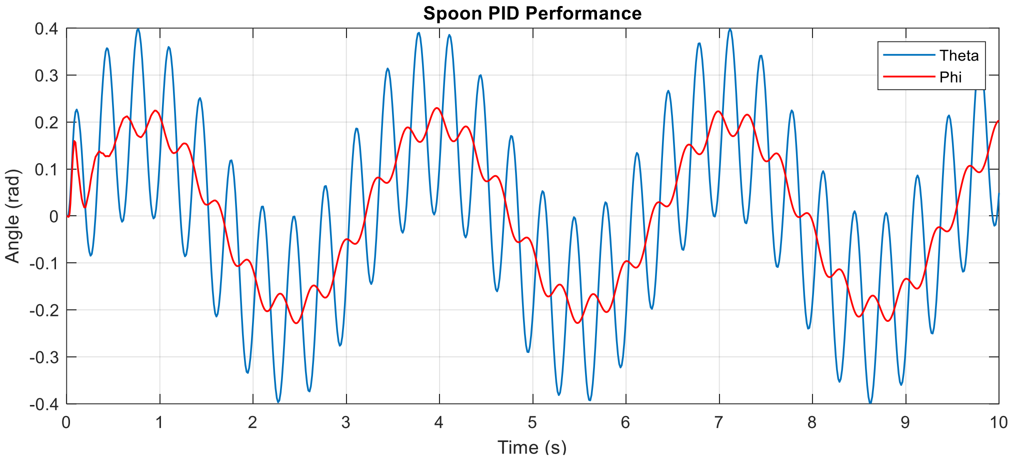


Figure 5.8 – Spoon PID Performance

B. PI CONTROL

Following this, the PI controller design was tested. Again, this was tuned until the best performance was seen, which can be seen in figure 5.9. As before the performance of the system is hindered by the cross-coupling that exists between the two systems, leading to small oscillations at the tremor frequency. The level of tremor removal was found to be approximately 82% or in decibels, -7.5 dB. These levels of attenuation along with the poor transient performance seen at the start of the simulation meant that this controller would not be used in the real system.

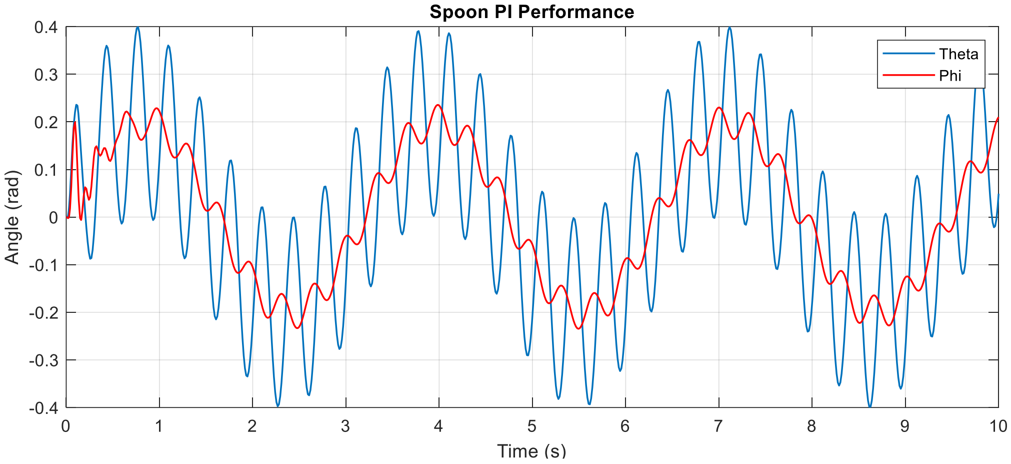


Figure 5.9 – Spoon PD Performance

C. PD CONTROL

The last iteration of this type of control system was the PD controller. The performance here was noticeably better than both of counterparts, which is shown below. With this controller the best level of attenuation of tremor movement was found to be -9.488 dB or approximately 88.75% removal of the tremor movement. This gives better performance than both the GYENNO and Liftware devices and should be easier to implement on Arduino controller, so it was clear that this would also be tested in the physical simulations.

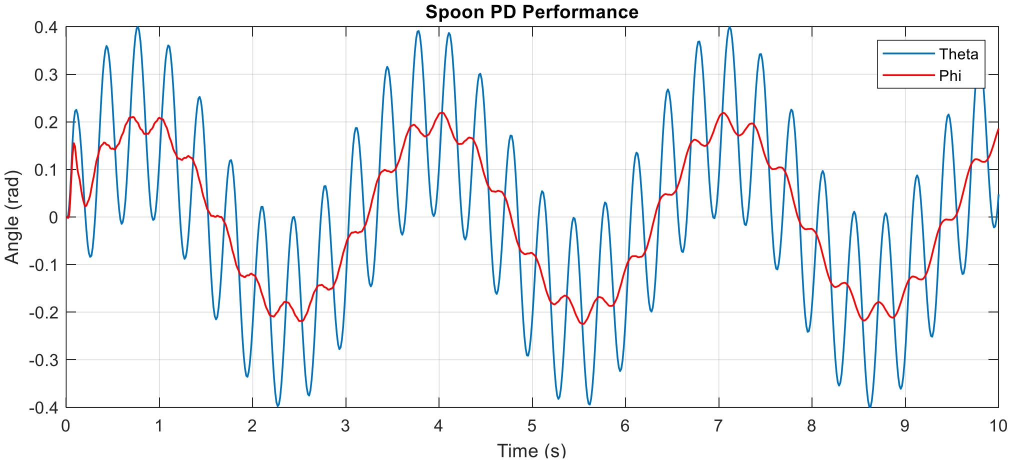


Figure 5.10 – Spoon PD Performance

D. Phase Lead Compensator

The last controller simulated was the Phase lead compensator. In this case the transfer function of the controller was tuned by increasing the gain to 5000 and changing the positions of the pole and zero to 0.1 and 0.3 respectively. The performance seen in these tests was the best so far with 90% removal of tremor which is equivalent to an attenuation of -10 dB. It was decided that this controller would be the primary system tested in the physical simulations as it gave the best performance. Furthermore, it would be easiest to implement as the matched pole zero method would be used rather than the integral and derivative approximations used in the PID and PD controllers.

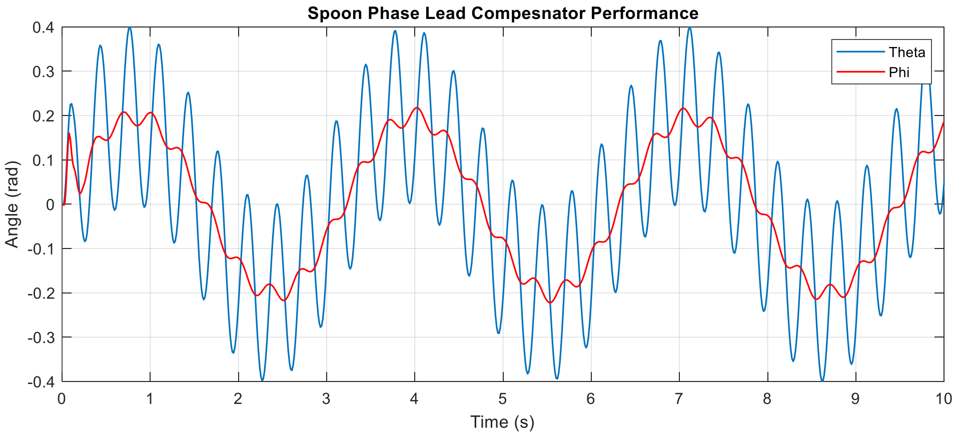


Figure 5.11 – Spoon Compensator Performance

5.2. Physical Simulation Results

Due to unforeseen circumstances our time working on this project was cut short. This meant that the system did not fully come to realisation in the physical apparatus. For this reason, results from the physical test system were not obtained. A lot of progress had been made, however, and a number of observations were made in the time spent implementing the control system on the physical apparatus. The process began by controlling the platform as this would form the basis for the testing to be carried out on the device. Initially, an attempt was made at using the PID controller to controller which was not successful as the platform would not settle in a position and would instead continue to oscillate ununiformly about the setpoint. Following this, the phase lead compensator was implemented. This gave much better behaviour and a sinusoidal setpoint with high and low frequency components was tracked adequately. Once this was achieved, work began on the spoon controller. With a setpoint of 0 rad set, the spoon did not settle as intended and instead oscillated uncontrollably. After some inspection it was found that a possible malfunction in the IMU may have become present giving the wrong angle readings. Unfortunately, we did not get to work on the device anymore after this.

6. DISCUSSIONS

The results outlined in the previous section of this report show several things. Firstly, the use of both types of controller that were investigated gave reasonable performance. In relation to the platform control, the PID and phase lead compensator offered near ideal behaviour. After some tuning in each case, the platform tracked the setpoint signal in a perfectly satisfactory manner. Knowing this, it was not surprising that the phase lead compensator worked so well in regard to the physical system. The platform showed oscillatory behaviour that was characterised by a high frequency component and low frequency component. In a test carried out using the oscilloscope to measure the platform potentiometer voltage, the high and low frequencies were measured to be 18.8 rads^{-1} and 2 rads^{-1} respectively. The reason that the PID controller didn't work in this instance could have been for a number of reasons. For one, the integration and differentiation functions may not have been accurate. More likely, it was due to wrong gain values used. Initially it was thought that the gains calculated in Simulink were too large and that this was due to a mistake in the model. After later investigation it was found that these gains were correct, considering that the input was a small value in radians and the output was a large value that was to be divided by 51 to give a voltage under 5 V. It is worth noting here that the use of the Ziegler Nichols method for tuning is questionable. Although it did give a good starting point, in both the spoon and platform the gain values were changed a lot during simulations.

The performance of the controllers designed were less effective when it came to spoon control system. While the system gave reasonable results with more than 85% of tremor being removed in the simulations, this was not as good as expected. In performing simulations like these, many real-world effects are not considered, and systems are greatly simplified. For one, the filter used in these tests showed near to ideal behaviour which could not be expected if it were to be implemented in the physical system. These simulations should have shown close to perfect performance so that the real-world implementation would have been as good as possible. In each of the simulations looking at the spoon controller performance a certain amount of tremor was seen at the output. This was not because of poor signal processing at the filter but actually due to the reactive torque present within the spoon motor. A back EMF seems to be generated in the motor due to the movement of the platform, which is in turn, causes the spoon to move. This effect was overlooked in designing of the spoon control system when it was assumed the platform would be stationary and therefore would not cause this reactive torque at the spoon. While the performance was hindered by this, the relative effectiveness of each controller can still be seen. It was clear that the phase lead compensator was the better of the designs as it gave the best attenuation of the high frequency. As well as this it would likely have been the most straight forward to implement, which seemed to be the case regarding the platform control system. In relation to the PID variety of controllers, the presence of the integral functions seemed to impede the control as the PI and PID controllers gave the poorest performance in simulations. The better behaviour of the PD controller can be explained by considering its tranfer function as shown below:

$$C(s) = \frac{5622\mathrm{z} - 2222}{z} = 5622\left(\frac{\mathrm{z} - 0.395}{z}\right)$$

Here it can be seen that the PD controller is effectively a phase lead compensator with a gain of 5000, and a pole and zero at 0 and 0.395 respectively.

Finally, the filter proved to be an equally important part of the design process. The notch filter gives superior performance over the low pass filters tested, but it is limited to a single frequency. Both low pass filters fall down in effective attenuation of the 18.8 rads^{-1} component but are not limited to a singular frequency like the notch filter is. As well as this an increase in tremor frequency means that the removal of tremor motion is improved as shown in appendices 5 and 6, where the frequency has been increased to 25 rads^{-1} in each. From these graphs it can also be seen that the Chebyshev filter seems to be better than the Butterworth as the high frequency is attenuated more when it is used. This is due to the steeper roll-off in its magnitude characteristic. For this reason, the Chebyshev filter seems like a better choice than the simple notch filter, if the spoon control system was improved than the small amount of tremor introduced by using this filter would have been acceptable.

7. Suggestion for Future Work

Although this project did not fully come to fruition, there is still a lot of work that was carried out, that can be built upon in future. The physical system that was created seemed to work very well. It was relatively straight forward to model and gave a good test bench to use the controllers with. The system mechanics equations worked well, as did the Simulink model. Some improvement could be made on the control system design, however. A better transfer function should be derived that will give a more accurate representation of the system to be controlled, thus leading to an improved controller algorithm. Another option would be to use a second form of control to remove the tremor seen on the spoon. Feed-forward control could be a solution to this. By passing the platform angle signal through a derivative function, followed by a high pass filter, the effects of the platform movement seen on the spoon could be nulled.

With regards the filtering there several possible alternatives. One could try to design a better low pass filter that would ensure no phase shift at low frequencies and full attenuation of tremor.

An easier option may to design a notch filter than adjusts its reject frequency by constantly monitoring the frequency of the tremor. This could be achieved by measuring time between peaks on the angle signal and then using this to find the tremor frequency.

A point that could be improved is the sampling rate of the controller. This was limited by the method used in calibrating the IMU sensor. A better means of retrieving data from the Arduino Uno should be researched. One option may be to increase the baud rate to allow faster communication between the device and serial monitor. Alternatively, a microcontroller with a larger memory could be used, so that the values can be stored and then transferred to the computer after the test run. As can be seen here this project has achieved quite a lot but still offers some scope for improvement.

8. Ethics and Safety

Ethics, health and safety are topics that should always be considered while practicing as an engineer. It is important to give attention to these aspects as they are often overlooked and without due care can lead to serious injury. There is a certain amount of responsibility placed on engineers in that they are the ones who are intrusted to design safe, user friendly systems that should not lead to injury under any circumstances. Regarding this project, care was taken to design a system that would not cause any harm to the user. The device that was designed will be used on a daily basis and will be in the hand of its user for long periods of time. For this reason, any risks associated with it should be minimal. The device should operate at low voltages and currents to prevent electrocution, low torque motors should be utilised to minimise possible harm in the case of malfunction and it must be waterproof as it will be washed and used with food regularly.

Health and safety had to be considered when carrying out the work for this project as well. Correct procedure was used when operating the testing equipment and any risks that were present were removed. This was achieved by carrying out a risk assessment and analysis of where problems may arise. This is shown in Appendix 9 of this report.

The risks that are associated with moving parts include loose hair and clothing getting caught in motors, gearboxes and axles. As well as this fingers and hands may get caught between moving parts of the device. These risks were lowered by ensuring loose clothing and hair were tied back and that participants stood away from testing equipment when it was operating. Stops were also placed on the device to prevent it from spinning uncontrollably.

Electronic components brought hazards in that they could overheat or operate at voltages and currents that were too high which could potentially injure people working with the equipment. To prevent this, protected power supplies were used, and current limits were set.

The cables and wiring used to power and measure the physical system may also cause harm in that they could be tripped over thus causing the test equipment to fall off the work bench. The solution to this was to ensure that no cables were ran across the floor, or between workbenches and that cabling and wires were shortened where possible.

Finally, there was risks associated with the heavy equipment that was used in the testing process. These could move and fall off the workbench due to vibrations caused by the moving parts of the equipment. This was prevented by ensuring all individuals stood away from the equipment when it was operating and by placing the moving apparatus at the centre of the workbench.

9. Conclusions

This report investigated a means of vibration rejection and stabilisation of a smart eating for people suffering with debilitating movement disorders such as PD and ET. The project began by researching the problem itself and investigating pre-existing solutions. Once an idea of the issue was grasped, a plan was made, and a physical test rig was devised that would mimic a tremoring hand. This gave a platform that the stabilisation system could be tested on. Following this the mechanics of the system were modelled, controllers were designed, and simulations were carried out to give an idea of the performance expected. Finally, the control system was implemented in part on the physical system and results were gathered where possible, to give an idea of the effectiveness of the vibration rejection.

In conclusion, this project has given great insight into many aspects of electrical engineering. While the project has mostly been focused on control engineering it was not limited to this, it has also looked at areas such as embedded systems, signal processing, mechanical engineering and coding. The quality work that was carried out throughout the course of the project meant that good performance was likely to be achieved from our system. Although our time working on the project was limited, it still leaves plenty of positive results that can be built upon in future work.

BIBLIOGRAPHY

[1] A. O'Sullivan (April 2020), "Vibration Rejection and Stabilisation of a Smart Eating Device", Retrieved: 9 April 2020.

[2] P. Taylor (September 2019), "Latest Innovations in Medical Device Technologies" [Online]. Available at: https://www.bccresearch.com/market-research/healthcare/innovations-in-medical-device-technologies.html. Retrieved: 1 April 2020.

[3] J. Meindl (March 1982), "Microelectronics and computers in medicine", Science (New York, N.Y.). 215. 792-7. 10.1126/science.7036345. Retrieved: 24 October 2019.

[4] J. H. McAuley, C. D. Marsden (August 2000), "Physiological and pathological tremors and rhythmic central motor control". Brain, Volume 123, Issue 8, August 2000, Pages 1545–1567. Retrieved: 1 April 2020.

[5] L. L. Rubchinsky, A. S. Kuznetsov, V. L. Wheelock, K. A. Sigvardt (April 2007), "Tremor". Scholarpedia, 2(10):1379. Retrieved: 2 April 2020.

[6] L. V. Kalia (19 April 2015), "Parkinson's disease". The Lancet, August 2015, Pages. 896-912. Retrieved: 6 November 2019.

[7] S. A. Isfahani, J. G. Farmer (July 2017), "Technology and Parkinson's Disease: New Solutions for Old Problems". [Online] Available at: https://practicalneurology.com/articles/2017-july-aug/ tech nology-and-parkinsons-disease-new-solutions-for-old-problems. Retrieved: 5 November 2019.

[8] Parkinson's Disease Foundation, "Statistics", Parkinson.org. [Online] Available at: www.parkinson.org/Understanding-Parkinsons/Statistics. Retrieved: 9 November 2019

[9] H. Rickards (16 February 2005), "Depression in neurological disorders: Parkinson's disease, multiple sclerosis, and stroke" Journal of Neurology, Neurosurgery & Psychiatry 2005;76:i48-i52. Retrieved: 2 April 2020.

[10] National Organisation for Rare Diseases (2015), "Essential Tremor", rarediseases.org. [Online]. Available at: https://rarediseases.org/rare-diseases/essential-tremor/. Retrieved: 2 April 2020.

[11] S. J. Groiss, L. Wojtecki, M. Südmeyer, and A. Schnitzler (November 2009), "Deep Brain Stimulation in Parkinson's Disease". Ther Adv Neurol Disord, 2009 Nov, Volume: 2(6), Pages: 20–28. Retrieved: 3 April 2020.

[12] M. A. Spindlerm, MD, J. Tarsy, MD (20 Novemeber 2019). "Initial pharmacologic treatment of Parkinson disease", uptodate.com [Online]. Avaiable: https://www.uptodate.com/contents/ initial-pharmacologic-treatment-of-parkinson-disease. Retrieved: 1 April 2020.

[13] "Steadiwear", steadiwear.com, 2020. [Online]. Available at: https://steadiwear.com/. Retrieved: 3 April 2020.

[14] "Liftware Steady", *Liftware.com, 2019.* [Online] Available at: https://www.liftware.com/steady/. Retrieved: 8 November 2019.

[15] "GYENNO Spoon", Gyenno.com, 2019. [Online]. Available at: https://www.gyenno.com/spoon. Retrieved: 8 November 2019.

[16] National Institutes of Health (17 March 2016), “A Spoon that Stabilizes Hand Tremors for Patients with Parkinson’s Disease and Essential Tremor”, nih.gov. [Online]. Available at: sbir.nih. gov/statistics/success-stories/lift-labs. Retrieved: 4 April 2020.

[17] A. Algoz, B. A. Hasnain (June 2018), “A control system for a 3-axis camera stablizer”. Retrieved: 4 April 2020.

[18] R. J. Rajesh and P. Kavitha (September 2015), "Camera gimbal stabilization using conventional PID controller and evolutionary algorithms," 2015 International Conference on Computer, Communication and Control (IC4). Retrieved: 4 April 2020.

[19] *"*Steadicam*"* tiffen.com*, 2019.* [Online]. Available at: https://tiffen.com/pages/steadicam. Retrieved: 9 November 2019.

[20] P. Saxena, R. V. Patel (November 2013), “An active handheld device for compensation of physiological tremor using an ionic polymer metallic composite actuator”. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. Retrieved: 4 April 2020.

[21] R. Shanahan, E. Long (April 2019), “Vibration Supression in "Smart Spoon" using Active Disturbance Rejection Control”. Retrieved: 3 November 2019.

[22] “PID Controller”, Wikipedia.org, 2020. Available at: en.wikipedia.org/wiki/PID_controller. Retrieved: 5 April 2020.

[23] “PID Theory Explained” ni.com*, 2019.* [Online]. Available at: http://www.ni.com/en-ie/innovations/white-papers/06/pid-theory-explained.html. Retrieved: 4 April 2020.

[24] G. Lightbody (2018), “Control Engineering I, Lecture Notes”. Retrieved: 1 September 2018.

[25] “Control”, freestudy.co.uk, 2020. [Online]. Available at: www.freestudy.co.uk/control/t8.pdf. Retrieved: 5 April 2020.

[26] “Arduino Motor Shield Rev3”, Arduino.cc, 2020. [Online]. Available at: https://store.arduino.cc/arduino-motor-shield-rev3. Received: 8 April 2020.

[27] “Arduino Uno Rev3”, Arduino.cc, 2020. [Online]. Available at: https://store.arduino.cc/arduino-Uno-rev3. Received: 8 April 2020.

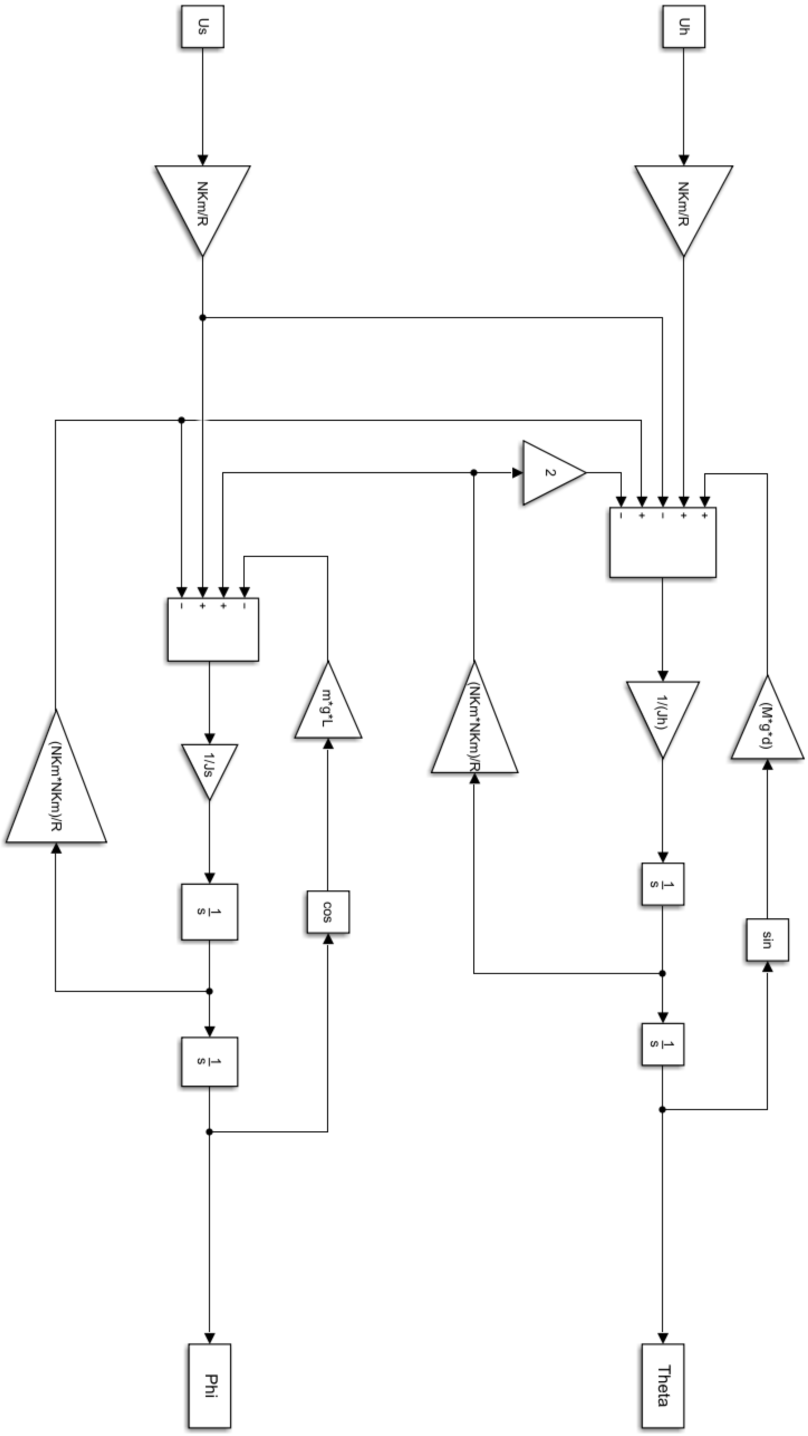
[28] J. Watson (2020), “MEMS Gyroscope Provides Precision Inertial Sensing in Harsh, High Temperature Environments”, Analog.com. [Online]. Available at: www.analog.com/ en/technical-articles/mems-gyroscope-provides-precision-inertial-sensing.html. Retrieved: 6 April 2020.

[29] G. Lightbody (2019), “Control Engineering II, Lecture Notes”. Retrieved: 1 September 2019.

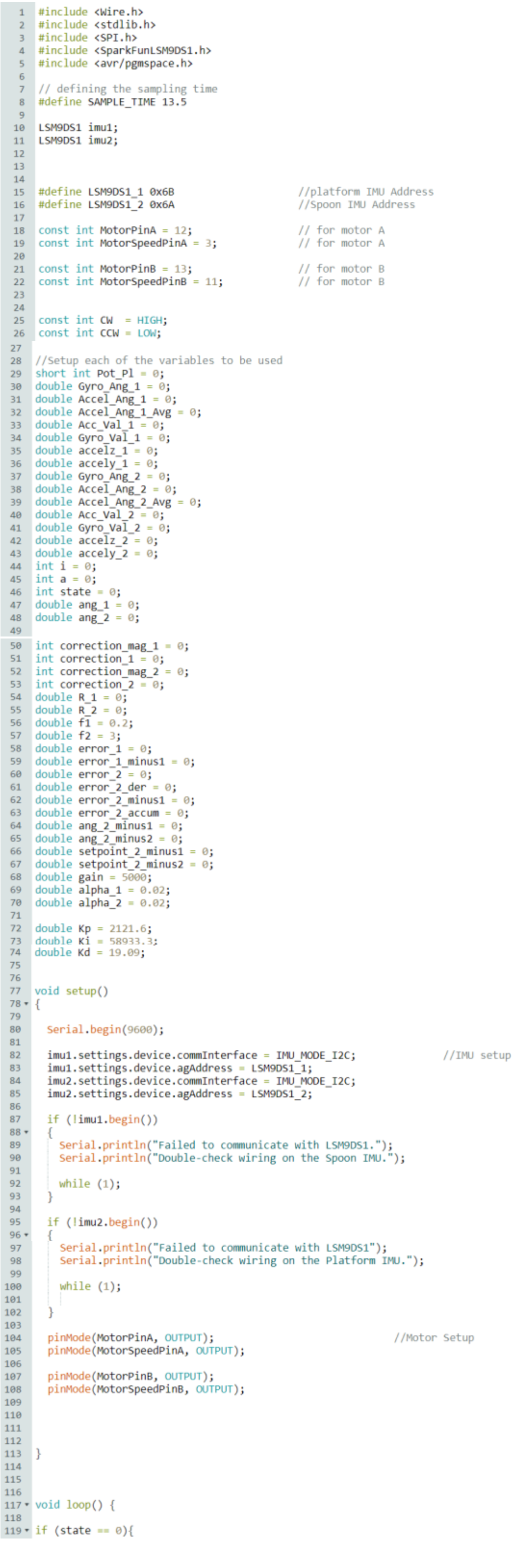
[30] R. Kavanagh (2018), “Signal Processing, Lecture Notes”. Retrieved: 1 September 2018.

Appendices

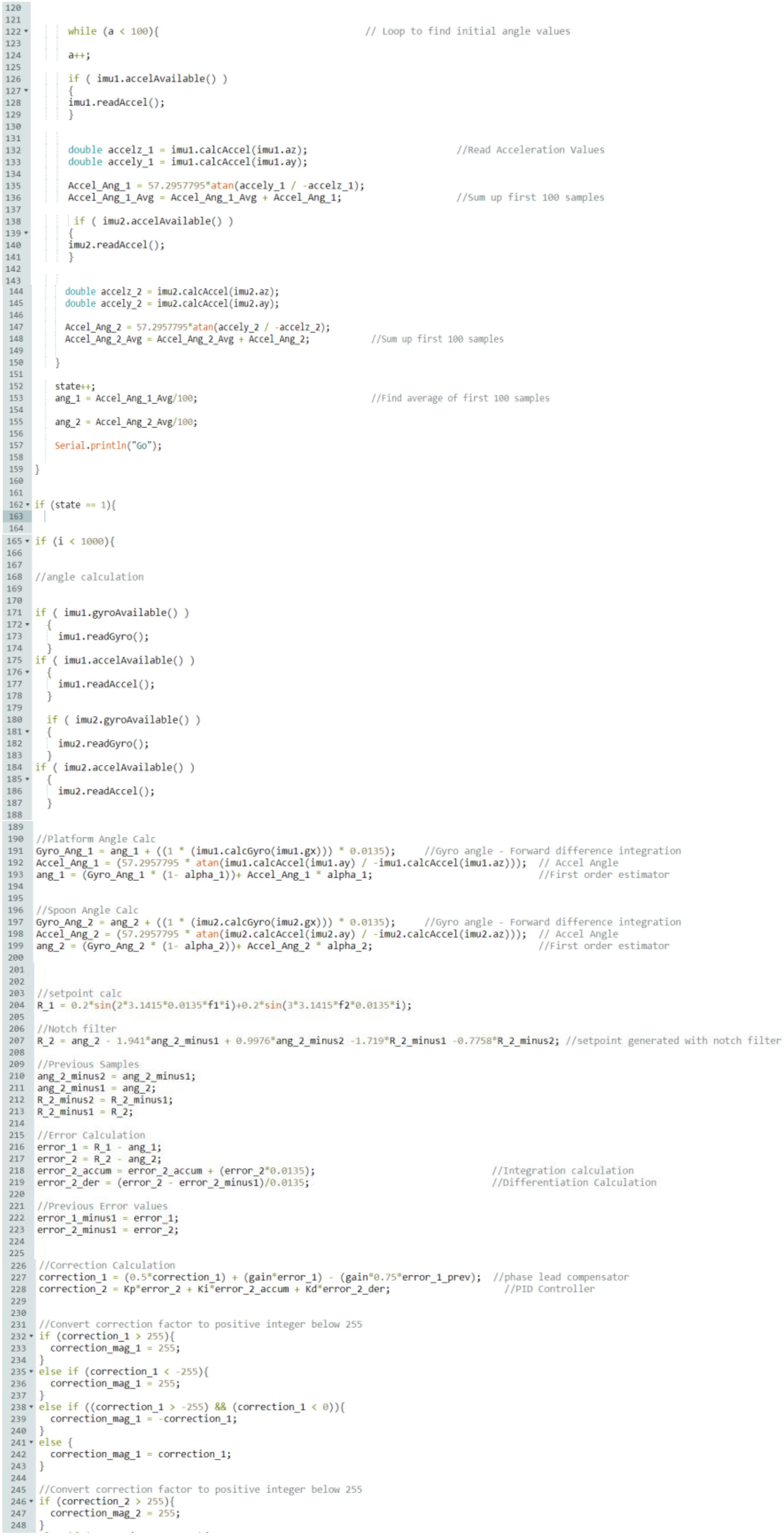
Appendix 1 – Simulink model of system mechanics:



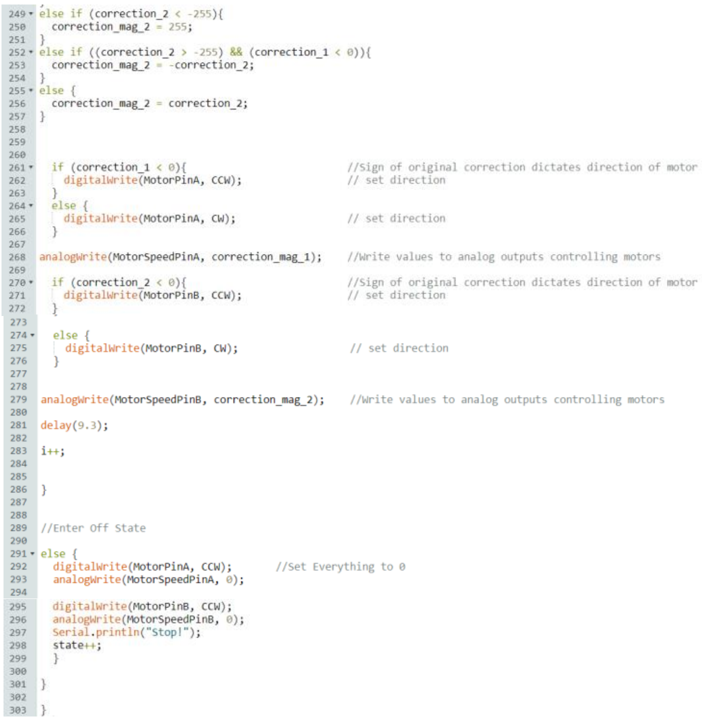
Appendix 2 – Arduino Code for full control System



```
#include <Wire.h>
#include <stdlib.h>
#include <SPI.h>
#include <SparkFunLSM9DS1.h>
#include <avr/pgmspace.h>

// defining the sampling time
#define SAMPLE_TIME 13.5

LSM9DS1 imu1;
LSM9DS1 imu2;



#define LSM9DS1_1 0x6B                  //platform IMU Address
#define LSM9DS1_2 0x6A                  //Spoon IMU Address

const int MotorPinA = 12;               // for motor A
const int MotorSpeedPinA = 3;           // for motor A

const int MotorPinB = 13;               // for motor B
const int MotorSpeedPinB = 11;          // for motor B


const int CW  = HIGH;
const int CCW = LOW;

//Setup each of the variables to be used
short int Pot_Pl = 0;
double Gyro_Ang_1 = 0;
double Accel_Ang_1 = 0;
double Accel_Ang_1_Avg = 0;
double Acc_Val_1 = 0;
double Gyro_Val_1 = 0;
double accelz_1 = 0;
double accely_1 = 0;
double Gyro_Ang_2 = 0;
double Accel_Ang_2 = 0;
double Accel_Ang_2_Avg = 0;
double Acc_Val_2 = 0;
double Gyro_Val_2 = 0;
double accelz_2 = 0;
double accely_2 = 0;
int i = 0;
int a = 0;
int state = 0;
double ang_1 = 0;
double ang_2 = 0;

int correction_mag_1 = 0;
int correction_1 = 0;
int correction_mag_2 = 0;
int correction_2 = 0;
double R_1 = 0;
double R_2 = 0;
double f1 = 0.2;
double f2 = 3;
double error_1 = 0;
double error_1_minus1 = 0;
double error_2 = 0;
double error_2_der = 0;
double error_2_minus1 = 0;
double error_2_accum = 0;
double ang_2_minus1 = 0;
double ang_2_minus2 = 0;
double setpoint_2_minus1 = 0;
double setpoint_2_minus2 = 0;
double gain = 5000;
double alpha_1 = 0.02;
double alpha_2 = 0.02;

double Kp = 2121.6;
double Ki = 58933.3;
double Kd = 19.09;


void setup()
{

  Serial.begin(9600);

  imu1.settings.device.commInterface = IMU_MODE_I2C;              //IMU setup
  imu1.settings.device.agAddress = LSM9DS1_1;
  imu2.settings.device.commInterface = IMU_MODE_I2C;
  imu2.settings.device.agAddress = LSM9DS1_2;

  if (!imu1.begin())
  {
    Serial.println("Failed to communicate with LSM9DS1.");
    Serial.println("Double-check wiring on the Spoon IMU.");

    while (1);
  }

  if (!imu2.begin())
  {
    Serial.println("Failed to communicate with LSM9DS1");
    Serial.println("Double-check wiring on the Platform IMU.");

    while (1);

  }

  pinMode(MotorPinA, OUTPUT);                           //Motor Setup
  pinMode(MotorSpeedPinA, OUTPUT);

  pinMode(MotorPinB, OUTPUT);
  pinMode(MotorSpeedPinB, OUTPUT);




}



void loop() {

if (state == 0){
```

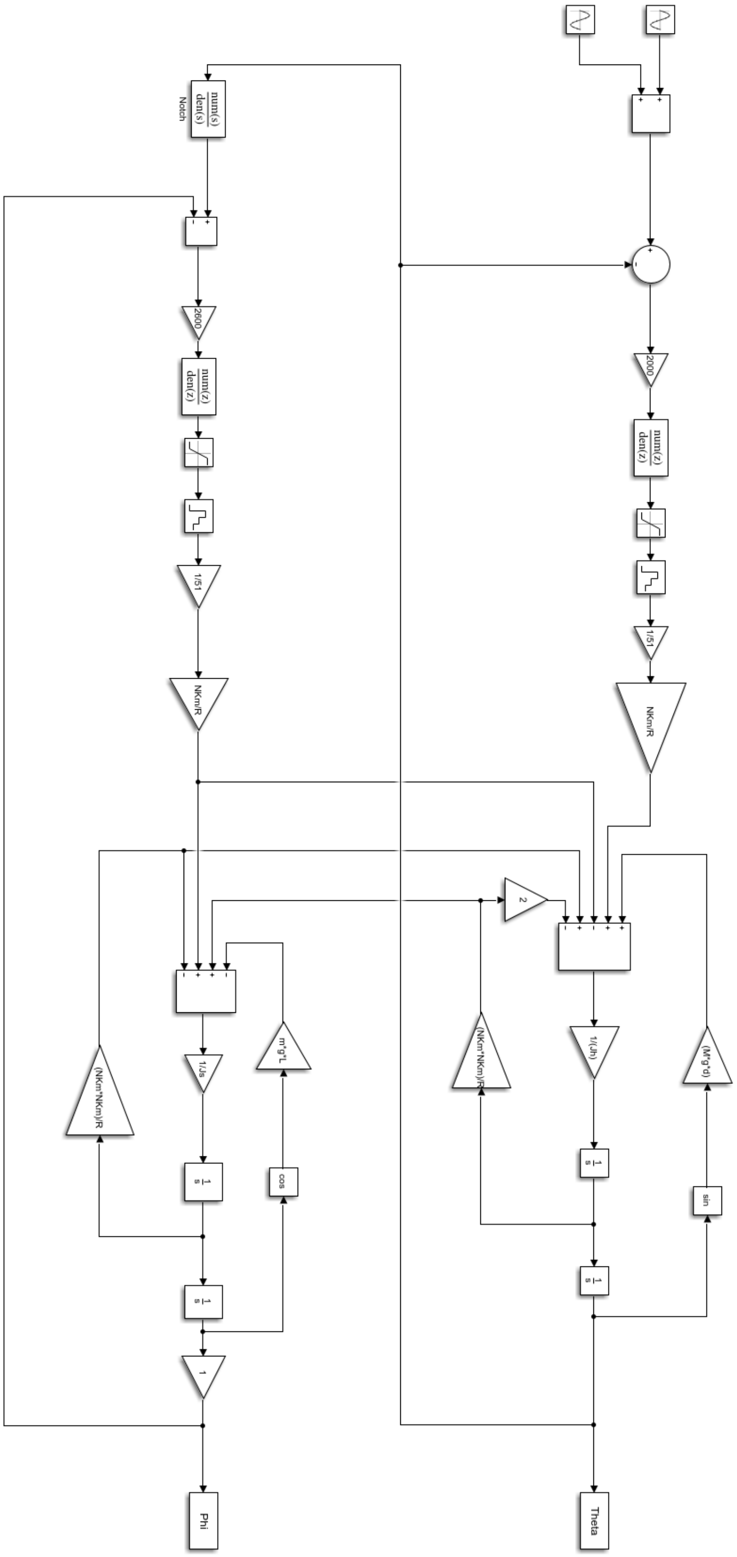


```


      while (a < 100){                                     // Loop to find initial angle values

      a++;

      if ( imu1.accelAvailable() )
      {
      imu1.readAccel();
      }


      double accelz_1 = imu1.calcAccel(imu1.az);                          //Read Acceleration Values
      double accely_1 = imu1.calcAccel(imu1.ay);

      Accel_Ang_1 = 57.2957795*atan(accely_1 / -accelz_1);
      Accel_Ang_1_Avg = Accel_Ang_1_Avg + Accel_Ang_1;                     //Sum up first 100 samples

       if ( imu2.accelAvailable() )
      {
      imu2.readAccel();
      }


     double accelz_2 = imu2.calcAccel(imu2.az);
     double accely_2 = imu2.calcAccel(imu2.ay);

     Accel_Ang_2 = 57.2957795*atan(accely_2 / -accelz_2);
     Accel_Ang_2_Avg = Accel_Ang_2_Avg + Accel_Ang_2;           //Sum up first 100 samples

    }

    state++;
    ang_1 = Accel_Ang_1_Avg/100;                               //Find average of first 100 samples

    ang_2 = Accel_Ang_2_Avg/100;

    Serial.println("Go");

}


if (state == 1){


if (i < 1000){


//angle calculation


if ( imu1.gyroAvailable() )
  {
   imu1.readGyro();
  }
if ( imu1.accelAvailable() )
  {
   imu1.readAccel();
  }

  if ( imu2.gyroAvailable() )
  {
   imu2.readGyro();
  }
if ( imu2.accelAvailable() )
  {
   imu2.readAccel();
  }


//Platform Angle Calc
Gyro_Ang_1 = ang_1 + ((1 * (imu1.calcGyro(imu1.gx))) * 0.0135);     //Gyro angle - Forward difference integration
Accel_Ang_1 = (57.2957795 * atan(imu1.calcAccel(imu1.ay) / -imu1.calcAccel(imu1.az)));  // Accel Angle
ang_1 = (Gyro_Ang_1 * (1- alpha_1))+ Accel_Ang_1 * alpha_1;                             //First order estimator


//Spoon Angle Calc
Gyro_Ang_2 = ang_2 + ((1 * (imu2.calcGyro(imu2.gx))) * 0.0135);     //Gyro angle - Forward difference integration
Accel_Ang_2 = (57.2957795 * atan(imu2.calcAccel(imu2.ay) / -imu2.calcAccel(imu2.az)));  // Accel Angle
ang_2 = (Gyro_Ang_2 * (1- alpha_2))+ Accel_Ang_2 * alpha_2;                             //First order estimator



//setpoint calc
R_1 = 0.2*sin(2*3.1415*0.0135*f1*i)+0.2*sin(3*3.1415*f2*0.0135*i);

//Notch filter
R_2 = ang_2 - 1.941*ang_2_minus1 + 0.9976*ang_2_minus2 -1.719*R_2_minus1 -0.7758*R_2_minus2; //setpoint generated with notch filter

//Previous Samples
ang_2_minus2 = ang_2_minus1;
ang_2_minus1 = ang_2;
R_2_minus2 = R_2_minus1;
R_2_minus1 = R_2;

//Error Calculation
error_1 = R_1 - ang_1;
error_2 = R_2 - ang_2;
error_2_accum = error_2_accum + (error_2*0.0135);                              //Integration calculation
error_2_der = (error_2 - error_2_minus1)/0.0135;                               //Differentiation Calculation

//Previous Error values
error_1_minus1 = error_1;
error_2_minus1 = error_2;


//Correction Calculation
correction_1 = (0.5*correction_1) + (gain*error_1) - (gain*0.75*error_1_prev);  //phase lead compensator
correction_2 = Kp*error_2 + Ki*error_2_accum + Kd*error_2_der;                    //PID Controller


//Convert correction factor to positive integer below 255
if (correction_1 > 255){
  correction_mag_1 = 255;
}
else if (correction_1 < -255){
  correction_mag_1 = 255;
}
else if ((correction_1 > -255) && (correction_1 < 0)){
  correction_mag_1 = -correction_1;
}
else {
  correction_mag_1 = correction_1;
}

//Convert correction factor to positive integer below 255
if (correction_2 > 255){
  correction_mag_2 = 255;
}
```

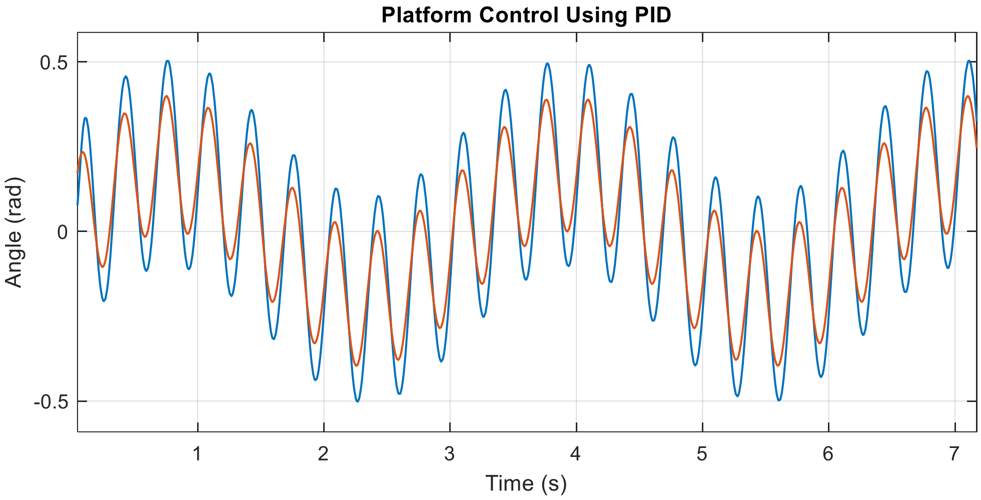


```
else if (correction_2 < -255){
  correction_mag_2 = 255;
}
else if ((correction_2 > -255) && (correction_1 < 0)){
  correction_mag_2 = -correction_2;
}
else {
  correction_mag_2 = correction_2;
}



  if (correction_1 < 0){                          //Sign of original correction dictates direction of motor
    digitalWrite(MotorPinA, CCW);                 // set direction
  }
  else {
    digitalWrite(MotorPinA, CW);                  // set direction
  }

analogWrite(MotorSpeedPinA, correction_mag_1);    //Write values to analog outputs controlling motors

  if (correction_2 < 0){                          //Sign of original correction dictates direction of motor
    digitalWrite(MotorPinB, CCW);                 // set direction
  }

  else {
    digitalWrite(MotorPinB, CW);                  // set direction
  }


analogWrite(MotorSpeedPinB, correction_mag_2);    //Write values to analog outputs controlling motors

delay(9.3);

i++;


}


//Enter Off State

else {
  digitalWrite(MotorPinA, CCW);       //Set Everything to 0
  analogWrite(MotorSpeedPinA, 0);

  digitalWrite(MotorPinB, CCW);
  analogWrite(MotorSpeedPinB, 0);
  Serial.println("Stop!");
  state++;
  }

}

}
```

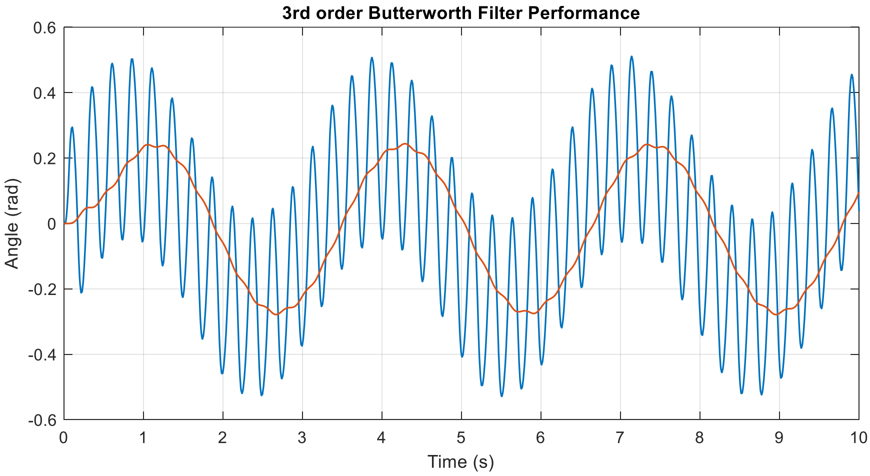
Appendix 3: Full Simulink Model:



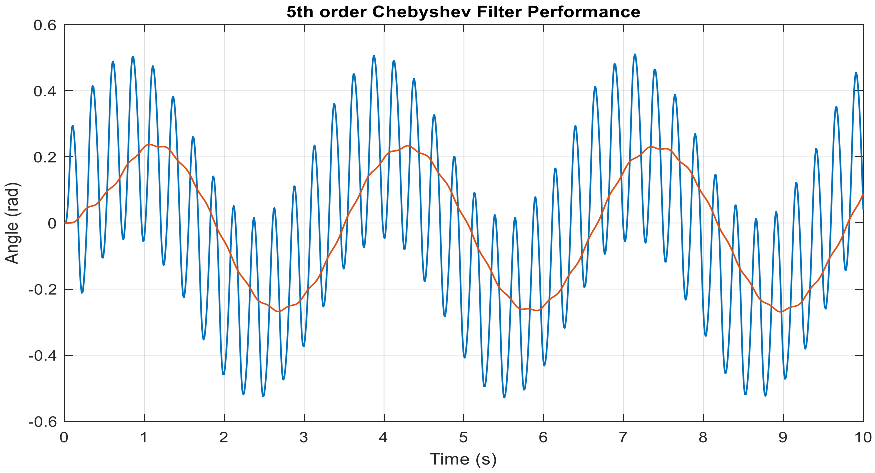
Appendix 4 - Platform control using PID controller (original):



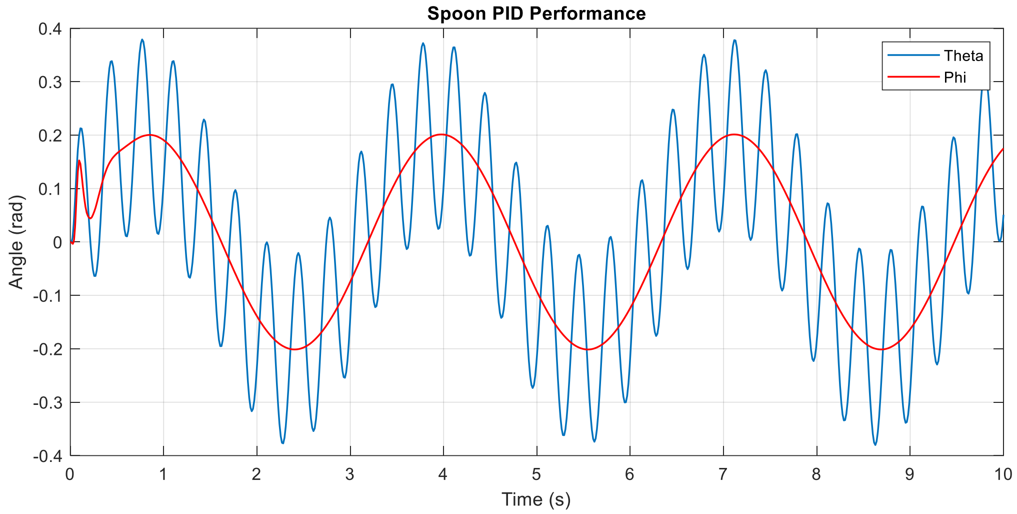
Appendix 5 – Butterworth Filter performance with tremor of 25 rads^{-1}:



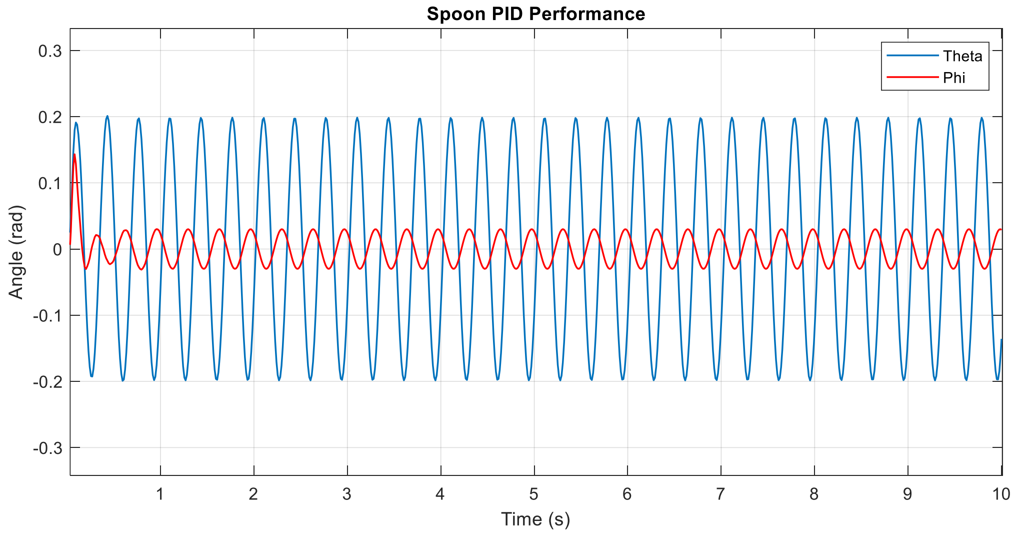
Appendix 6 – Chebyshev Filter performance with tremor of 25 rads^{-1}



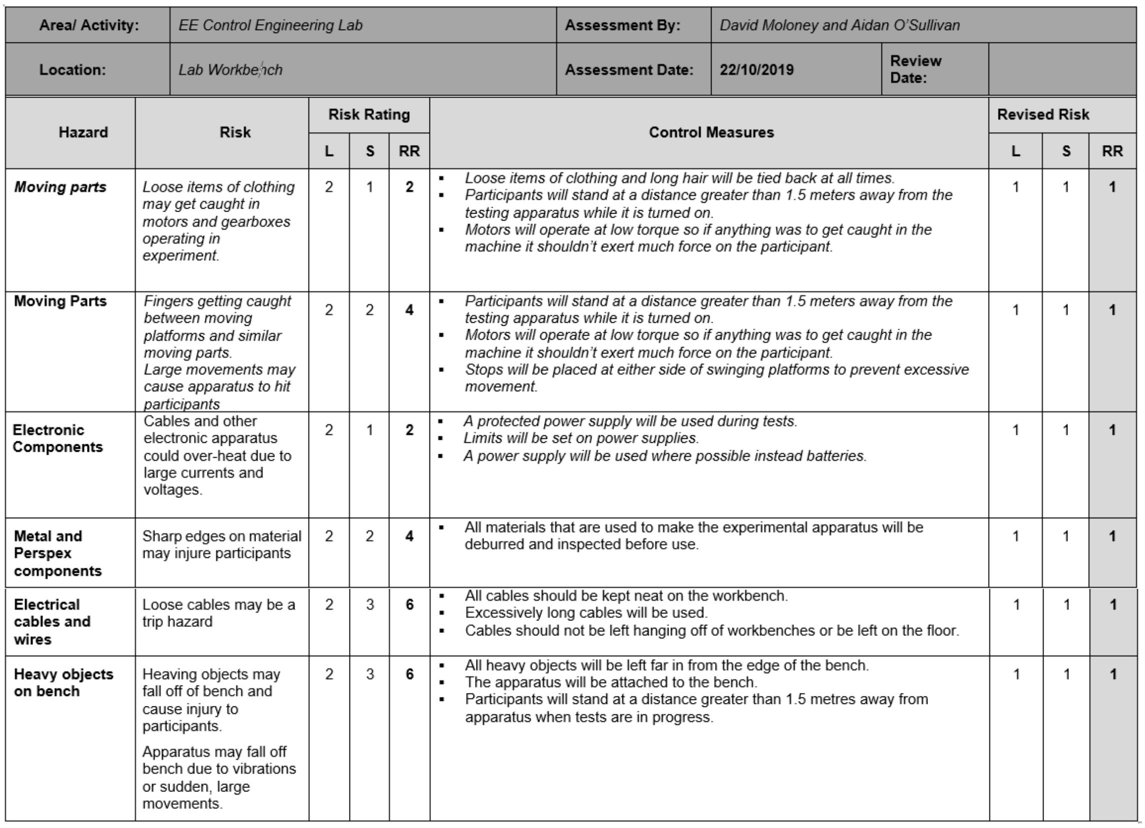
Appendix 7 – Spoon PID performance without platform effects:



Appendix 8 – Spoon PID performance with tremor frequencies:



Appendix 9 – Risk Assessment:



Area/ Activity:	*EE Control Engineering Lab*	**Assessment By:**	*David Moloney and Aidan O'Sullivan*		
Location:	*Lab Workbench*	**Assessment Date:**	**22/10/2019**	**Review Date:**	

Hazard	**Risk**	**Risk Rating**			**Control Measures**	**Revised Risk**		
		L	**S**	**RR**		**L**	**S**	**RR**
Moving parts	*Loose items of clothing may get caught in motors and gearboxes operating in experiment.*	2	1	**2**	▪ *Loose items of clothing and long hair will be tied back at all times.* ▪ *Participants will stand at a distance greater than 1.5 meters away from the testing apparatus while it is turned on.* ▪ *Motors will operate at low torque so if anything was to get caught in the machine it shouldn't exert much force on the participant.*	1	1	**1**
Moving Parts	*Fingers getting caught between moving platforms and similar moving parts.* *Large movements may cause apparatus to hit participants*	2	2	**4**	▪ *Participants will stand at a distance greater than 1.5 meters away from the testing apparatus while it is turned on.* ▪ *Motors will operate at low torque so if anything was to get caught in the machine it shouldn't exert much force on the participant.* ▪ *Stops will be placed at either side of swinging platforms to prevent excessive movement.*	1	1	**1**
Electronic Components	Cables and other electronic apparatus could over-heat due to large currents and voltages.	2	1	**2**	▪ *A protected power supply will be used during tests.* ▪ *Limits will be set on power supplies.* ▪ *A power supply will be used where possible instead batteries.*	1	1	**1**
Metal and Perspex components	Sharp edges on material may injure participants	2	2	**4**	▪ All materials that are used to make the experimental apparatus will be deburred and inspected before use.	1	1	**1**
Electrical cables and wires	Loose cables may be a trip hazard	2	3	**6**	▪ All cables should be kept neat on the workbench. ▪ Excessively long cables will be used. ▪ Cables should not be left hanging off of workbenches or be left on the floor.	1	1	**1**
Heavy objects on bench	Heaving objects may fall off of bench and cause injury to participants. Apparatus may fall off bench due to vibrations or sudden, large movements.	2	3	**6**	▪ All heavy objects will be left far in from the edge of the bench. ▪ The apparatus will be attached to the bench. ▪ Participants will stand at a distance greater than 1.5 metres away from apparatus when tests are in progress.	1	1	**1**

Appendix 10 – Logbook:

Week 1: 29th of September

Objectives:

- Complete ethics report.
- Organise weekly progress meeting with FYP supervisor.
- Research vibration reduction solutions and pre-existing solutions.
- Read report from last year's project.

Achievements:

- Read IEEE and Engineers Ireland Code of Ethics.
- Researched Sulfur Hexafluoride environmental problems and possible solutions.
- Organised weekly sync-up time with Gordon.
- Researched existing solutions and vibration reduction control systems.

Plan for the following week:

- Investigate different control methods.
- Complete further research of vibration reduction control.
- Research different motors for countering vibration.
- Read over third year notes.

Week 2: 6th of October

Objectives:

- Look at possible motors.
- Read over third year notes.
- Research different control systems.
- Meet up with Gordon and discuss the project.

Achievements:

- Discussed the advantages of using different kinds of motors and various means of delivering torque to the spoon for vibration cancellation.
- Discussed various other physical forms that could be used for the system.
- Investigated how to reduce mechanical issues such as cogging and back lash that arose in the last years project.
- Researched some more control systems and existing solutions.
- Read over third year notes.

Plan for the following week:

- Complete Risk Assessment Form
- Meet with Hillary Mansfield in the control lab.
- Complete more research on control systems, motors and MCU's.

Week 3: 13th of October

Objectives:

- Meet with Hillary in the control lab to discuss possible physical set up.
- Look at possible motors that can be used for the device.
- Decide on how to drive the motor e.g Driver card, Arduino Motor Shield.
- Complete risk assessment.

Achievements:

- Met Hillary and tested and discussed various motors and their suitability for controlling the device.
- Completed the risk assessment form by looking a laboratory set up and seeing what may be a risk. By grading each risk on its severity and likelihood, the risk rating could be calculated. If the risk was too high, steps were taken to lower it.

Plan for the following week:

- Model mechanics of system.
- Look the use of different motors; brushless and brushed motors. Look into Maxon motor and RS motor.
- Look at method of controlling motor; Arduino and Motor Shield, H-bridge, drive card.
- Try to set up the IMU.

Week 4: 21st of October

Objectives:

- Research and IMU and begin setting up.
- Begin taking acceleration and angle measurements from the IMU.
- Look at RS motors.
- Discuss alterations to be made to physical apparatus.

Achievements:

- Decided to use and I2C interface as it requires less wires.
- Connected the I2C hardware and downloaded the IMU Arduino libraries but had some trouble using the Arduino.
- Decided that two motors would be used for the set up, one for the control of the spoon, one for the control of the base.

Plan for the following week:

- Compare the Motor Shield specs to the voltage requirements of the available motors in the lab.
- Continue looking ta IMU and try to fix problems with Arduino.
- Take Physical testing apparatus to mechanical workshop and figure out how we can set up motors.

Week 5: 28th of October

Objectives:

- Begin writing the preliminary FYP report.
- Take device to mechanical workshop and see can they assemble the dual motor system we desire.
- Continue working with IMU

Achievements:

- Took prototype to mechanical workshop.
- Got IMU set up along with Arduino. Began reading values for Accelerometer, Gyroscope and Magnetometer. Calculated pitch, roll and heading using these values.
- Spoke to Gordon about the system equations and gained good understanding of them.

Plan for the following week:

- Collect prototype from mechanical workshop.
- Begin trying to connect and control the new motors with the Arduino motor shield.
- Begin writing the Preliminary report.

Week 6: 4th of November

Objectives:

- Collect prototype form workshop.
- Begin reading for preliminary report and start writing literary review.
- Meet with Gordon and discuss the physical set up and what should be looked at next.

Achievements:

- Collected prototype from workshop.
- Looked into using the motor shield in conjunction with Arduino.
- Spoke to Gordon about the physical set up and what this means for future work. As the system is now unstable, we must now create a control system for the motor operating the platform. This won't be very difficult and will be good practice for us.
- Began working on preliminary report.

Plan for the following week:

- Finish preliminary report.
- Look at controlling motor using motor shield.
- Look at controlling platform.
- Make a plan for the second semester using a Gantt chart.

Week 7: 11th of November

Objectives:

- Finish preliminary report.
- Meet with Gordon and discuss plan for foreseeable future.
- Plan work to be carried out.

Achievement:

- Finished preliminary report and submitted it.
- Met Gordon and discussed some of the important aspect of the report and how to plan work for the coming months.
- Created a Gantt chart with partner and divided up workload between us. Created contingency plans that would be used in the case of any setbacks.
- Set deadlines to ensure progress is made a good rate.

Plans for next week:

- Start looking at driver shield implementation.
- Do more research on possible control methods.

Week 8: 6th of January

Objectives:

- Implement the driver shield with DC motors.
- Begin work on Simulink model.

Achievement:

- Began Looking at root locus design. Research the process and revised notes from last semester.
- Started using Simulink to become familiar with the interface again.
- Generated some basic, first iterations of the Simulink model.
- Looked at the driver shield and researched how it was used to control DC motors

Plans for Next Week:

- Continue work on modelling of system mechanics.
- Create Simulink model.
- Look into root locus design methods.

Week 9: 13th of January

Objectives:

- Look at Simulink modelling.
- Finalise system mechanics equations.
- Speak to Gordon about possible control techniques.

Achievement:

- Derived system mechanics equations with Gordons help.
- Decided that two control systems would be used which both used an IMU. The platform angle would then be fed into a filter which would remove the high frequencies and this would become setpoint for the spoon system
- Made a plan for the foreseeable future with regards designing the control system.
- Began looking at creating a Simulink model for the system.

Plans for Next Week:

- Continue work on Simulink model.
- Calculate and gather constants associated with physical system to be used in modelling.

Week 10: 20th of January

Objectives:

- Calculate constants for modelling of system mechanics.
- Calculate moments of inertia associate with platform and spoon.
- Begin looking at the calibration of the IMU sensors.

Achievement:

- Worked on Simulink model representing the system using the system mechanics equations
- Found mechanical constants using Solidworks.
- Took measurements from physical system and created the design in the 3D CAD environment.
- Found position of centre of gravity, mass of system and moment of inertia about the platform axle.

Plans for Next Week:

- Complete the Simulink model and begin designing controllers.
- Derive transfer functions for use in controller design.
- Work on angle measurement using IMU.

Week 11 27th of January

Objectives:

- Implement the first order estimator to be used in measuring the angles of the platform and spoon.
- Begin planning presentation which is to be carried out in a few weeks

Achievement:

- Researched first order estimator to be used with the IMU.
- Researched the IMU itself and found out how acceleration and gyroscope reading are used to measure angles.

- Met with Gordon who explained how to calibrate IMU and what we should aim for with regards the presentation.

Plans for Next Week:

- Begin working on presentation.
- Calibrate the angle estimator by comparing IMU to potentiometer readings.

Week 12: 3rd of February

Objectives:

- Make a plan for the presentation and start working on.
- Calibrate estimator that uses IMU readings to generate angle.
- Set up potentiometer with voltage source so values between 0 and 5 V are outputted.

Achievement:

- Began working on presentation that is due next week, made a plan and broke up the work between Aidan and I.
- Calibrated potentiometer on platform to give voltages between 0 and 5V.
- Measured angles on platform and created an equation that would generate an angle using the voltage value.
- Looked at different methods of reading the potentiometer and IMU values so that IMU could be calibrated.

Plans for Next Week:

- Concentrate mostly on presentation.
- Prepare slideshow and script.
- Record IMU and potentiometer values to calibrate estimator.

Week 13: 10th of February

Objectives:

- Work on presentation. Make slideshow and script.
- Practice presentation with Aidan and look at possible questions.
- Finish calibrating IMU sensor.

Achievements:

- Continued working on Presentation.
- Made slideshow.
- Wrote script and learned it off.
- Met Gordon to see if he had any pointer.
- Practiced presentation multiple times.
- Carried out presentation.
- Imported data to from IMU and potentiometer readings to Matlab and calibrated estimator.

Plans for Next Week:

- Continue working on platform controller.

- Try to implement PI control of the system.

Week 20: 17th of February

Objectives:

- Implement PI control on the platform.
- Find dead bands associated with controllers and add this to code.
- Work on designing controllers using Simulink.

Achievements:

- Began implementing PI controller on platform.
- Used Euler's Forward difference method.
- Dead band size was found by balancing platform at level position and then incrementing bit until it moved.
- Found out that the serial monitor was using up a lot of the time during cycles.

Plans for Next Week:

- Figure out a means of lowering the sampling time of the controller.
- Recalibrate the IMU for new sampling time.
- Begin Work on phase lead compensator.

Week 21: 24th of February

Objectives:

- Design phase lead compensator to be used in platform control.
- Figure out a means of calibrating the IMU without printing values to serial monitor during cycle.

Achievements:

- Began looking at designing a phase lead compensator for platform.
- Generated a transfer function for the system and looked at root locus design.
- Figured out what assumptions could be made in finding this.
- Looked other methods of decreasing the sampling time of the controller.
- Tried to store values in an array and then print to the monitor but system memory was not big enough

Plans for Next Week:

- Recalibrate IMU estimator and finalise minimum sample time that can be used.
- Finish design of the phase lead compensator and implement compensator in the platform control system.

Week 22: 2nd of March

Objectives:

- Recalibrate IMU.
- Design Phase Lead Compensator.

- Simulate Phase lead compensator using Simulink.

Achievements

- Used Sisotool to figure out positions for phase lead compensator poles.
- Implemented compensator on Arduino using the matched pole zero method.
- Sinusoidal behaviour was observed on the platform.
- The IMU had to be recalibrated, after minimising the time to print to values to the serial monitor.
- Began work on spoon controller. Calibrated potentiometer using the voltage source and measuring the maximum angles that spoon could move to.
- Tried to set up two I2C communication channels.

Plans for Next Week:

- Find solution to I2C address problem.
- Implement spoon controller.
- Implement full control system.

Week 23: 9th of March

Objectives

- Get two IMUs reading data at the same time.
- Design Controller for spoon system.
- Implement this controller along with filter.
- Finish entire control system.

Achievements:

- Found out why the data from the I2C was being corrupted.
- Problem was fixed by changing the address of the second IMU.
- Looked at design phase lead compensator for the spoon using root locus plot.
- Derived a transfer function for the spoon system.
- College was closed down so our work on the device was finished prematurely.

Plans for Next Week:

- Finish Simulink model and design more types of filter.
- Begin Working on poster.

Week 24: 16th of March

Objectives:

- Finish Simulink model and similate entire control system.
- Start work on poster.
- Plan video.

Achievements:

- Continued designing spoon controller using Simulink model.
- Began looking at filters to be used in the system other than the notch filter.
- Began work on poster and video presentation.

Plans for Next Week:

- Finish poster.
- Make video presentation of poster.
- Continue modelling the System.

Week 25: 23rd of March

Objectives:

- Finish poster.
- Sync up with Aidan to decide on how to make video.
- Make video and submit.

Achievements:

- Continued working on poster.
- Planned the poster and video with Aidan.
- Wrote a script and recorded voice over clips.
- Used adobe video editor to make a video that showed off the poster effectively along with any other material that was relevant.

Plans for Next Week:

- Begin Work on final report.
- Finalise Simulink model and carry out simulations.
- Email Gordon.

Week 26: 30th of March

Objectives

- Make a plan for the report.
- Begin Writing the report.
- Finalise Simulation results.

Achievements.

- Began drafting the report for the project.
- Created a structure and decided what would be included.
- Finalised Simulink models to be used in the project report.

Plans for Next Week:

- Finish writing report.
- Format report.
- Proof read and ensure all information is included.

Week 27: 6th of April

Objectives:

- Finish writing the report.
- Proofread report and edit as required.

Achievements:

- Finished report writing.
- Read over the report and formatted it as per the specifications.
- Had to cut out a few sections that bringing the report of the 50 page mark
- Submitted the report.