

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

MSC CONTROL SYSTEMS

Construction and Control of a 2DOF Inverted Pendulum with Propellers and a Ground Robot

Author

Charlie Bell

Supervisor

Dr. Eric Kerrigan

September 2018

Acknowledgements

First and foremost I would like to thank Eric Kerrigan and Ian McInerney for supervising this project.

Secondly I would like to thank Francisco Grifo for the initial work on the project and Orsika Bjortar for technical advice throughout. Christian Porter deserves a mention for his mathematical expertise.

Additionally I want to acknowledge that none of this would be possible without the support from my parents, my close friend Sophia Kosmider and her mother Louise Alp.

I would like to mention Holly Wilson for being the God of all living things. Cara Ludlow for her inspirational music. Caterina Gioia for giving a goal to work towards. Ilayda Karadag for assistance with obtaining the correct parts.

Abstract

Analysis and hardware testing of a highly nonlinear quadrotor top inverted pendulum is performed. The system is a 6 degree of freedom inverted pendulum mounted with propellers. An Arduino Mega 2560 board is used as a microcontroller. The cart component is driven by differential drive motors. The motors receives signals from a Bluetooth module which communicates with an Android phone. Development of the Android application is also discussed.

An MPU6050 is used for angle readings of the quadrotor component; the MPU6050's onboard fusion and a complementary filter are compared. Proportional-integral-derivative feedback is used to stabilise the pendulum arm in the upright position. Details of the construction, including parts sourced and manufactured, are given throughout, alongside a description of the hardware assembly, software and subsystems. The feedback controller proves sufficient with stabilising the system as desired; and the MPU6050's onboard fusion outperforms the complementary filter.

Modelling and simulations are produced for future model based control.

List of Figures

1	Q-TIP concept design.	1
2	Hardware overview	4
3	Castor wheel on wooden frame attachment.	5
4	Cart and power supply	6
5	Previous quadrotor arm design. [1]	7
6	New quadrotor arm design.	7
7	MPU6050 base design and attachment.	8
8	IR distance sensor mount.	9
9	Reflective plate for IR distance sensor.	10
10	Base cone mount for reflective plate.	10
11	Generalised subsystem block diagram.	12
12	Ground motors and driver setup. (Adapted from [2] and [3])	13
13	Byte value to direction map.	14
14	HC-05 Bluetooth connection scheme. (Adapted from [4] and [5])	15
15	Android mobile app GUI.	15
16	IR distance sensor scheme. (Adapted from [6])	16
17	MPU6050 schematic. (Adapted from [7])	17
18	Complementary filter and unfiltered response.	18
19	Filtering method comparison.	19
20	ESC and quadrotor motors scheme. (Adapted from [8])	20
21	Coordinate systems and transformed frames.	23
22	Simulations of 1 DOF pendulum on 1 DOF cart.	25
23	Unforced response with constant yaw.	26
24	Unforced response with constant yaw rate.	27
25	Hardware response from large initial roll.	28
26	Hardware response from large initial pitch.	29
27	Hardware response from large initial roll and pitch.	30
28	Hardware response from equilibrium initial condition.	31

Contents

1	Introduction	1
1.1	Problem Overview	1
1.2	Project goals	1
1.3	Report structure	2
2	Background	3
2.1	Previous work on 3D inverted pendulums	3
2.2	Background on hardware modules	3
2.3	Useful software	3
3	Mechanical design and hardware assembly	4
3.1	Overview	4
3.2	Cart and wheels	5
3.3	Power supply	5
3.4	Quadrotor	7
3.5	Sensors	8
3.6	Safety	11
4	Subsystems and software	12
4.1	Overview	12
4.2	Ground motors	13
4.3	Wireless control and interface	14
4.4	Sensor readings	16
4.5	Filtering	18
4.6	ESCs and top motors	19
4.7	Feedback control	20
5	Hardware Operation	21
5.1	Operating procedure	21
5.2	Troubleshooting and common problems	21
6	Modelling and simulations	23
6.1	Modelling	23

6.2	Model verification by simulation: 1 DOF pendulum on 1 DOF cart	25
6.3	Model verification by simulation: 3 DOF pendulum	26
7	Hardware response	28
7.1	Roll error feedback performance	28
7.2	Pitch error feedback performance	29
7.3	Combined roll and pitch error feedback performance	30
7.4	Steady state tracking performance	31
8	Conclusions	32
8.1	Summary of Q-TIP	32
8.2	Future work	32

1 Introduction

1.1 Problem Overview

Quadrotor Top Inverted Pendulum (Q-TIP) is a 6 degrees of freedom (DOF) pendulum on a cart designed in [1]. Q-TIP is highly nonlinear, and so this thesis aims to develop a platform on which University students can gain practical experience with controller development in nonlinear systems.

This project details the steps taken in the construction, the control and the stabilisation of the entire Q-TIP system. Figure 1 illustrates the concept design of the system.

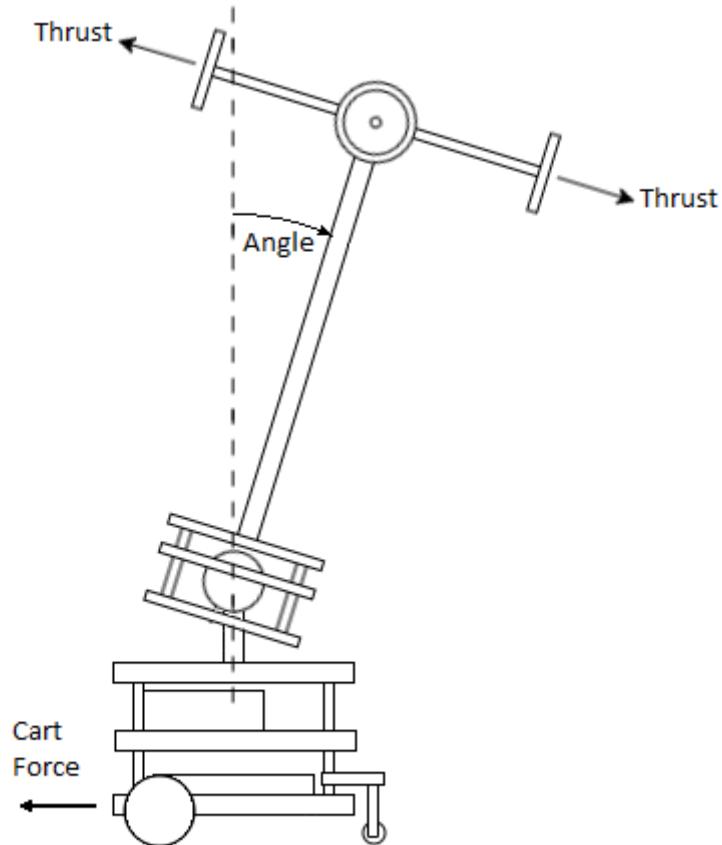


Figure 1: Q-TIP concept design.

The system consists of a pendulum and a cart connected by a 3 (DOF) ball joint. The cart moves around by instruction sent to the onboard microcontroller, by the operator, via a mobile application (app). The pendulum arm is then stabilised by feedback to four propulsion units. These units are placed laterally at the top of the pendulum: each unit consisting of a motor and a propeller.

1.2 Project goals

This project is defined by a series of aims and objectives.

Project aims:

- Finish construction of the robot, using a cart which had already been developed.
- Apply proportional-integral-derivative (PID) control to stabilise the pendulum arm.

- Drive the cart with some form of user interface.

Project objectives:

- Select and order essential parts.
- Design, 3D print and laser cut sensor mounts and other components.
- Assemble the mechanical structure.
- Connect hardware and wire subsystems.
- Read and process sensor data from the MPU6050 gyroscope and accelerometer.
- Read data from the infrared (IR) reflective distance sensors.
- Design a complementary filter to combine gyroscope and accelerometer readings.
- Read angle data from the MPU6050's Digital Motion Processor (DMP).
- Implement a stabilising PID controller for the pendulum arm.
- Develop an app and Arduino code to drive the cart motors.

1.3 Report structure

The report begins by providing background of the previous work done on Q-TIP in Section 2. Hardware, manufacturing and mechanical setup is detailed throughout Section 3. Subsystems and software are thoroughly documented in Section 4. A user guide is described in Section 5 to provide instructions on how to operate Q-TIP. Modelling and simulations of the system are carried out in Section 6. Results from hardware testing are analysed in Section 7. Finally Section 8 draws conclusions from the hardware, software and simulations.

2 Background

The inverted pendulum on a cart is a widely researched example of a control system; it is most commonly studied in a single plane, consisting of a single rotational DOF and translational DOF.

2.1 Previous work on 3D inverted pendulums

Uding originally designed and controlled a propeller mounted inverted pendulum on an omnidirectional ground robot [9]. The system consisted of 5 DOF, the cart making 3: longitudinal translation, transverse translation, and heading angle. The other 2 DOF came from the roll and pitch rotations of the pendulum arm. Both PID and LQR control were implemented to successfully stabilise the system, with PID performing the best.

Grifo selected, designed and manufactured the major components of Q-TIP detailed in his thesis [1]. He designed the overall general structure, selected materials, and dimensions. He developed the ball joint connecting the pendulum and cart to have a variable DOF, varying from 1 to 3. He also designed the mechanical parts, and selected the electronic components to be used in the system. Overall the system designed, totals 6 DOF. The cart Grifo designed is a two wheel drive, differential drive robot, constraining any transverse movement. Assembly of the system was not fully complete; Section 3 discusses the final assembly, as well as the parts sourced and manufactured.

2.2 Background on hardware modules

The HC-05 Bluetooth module, and the MD10C motor drivers have plenty of documentation available online, and user guides on how to set them up [4] [2]. The electronic speed controllers (ESCs) used did not have official documentation, but it was simple to figure out the basic calibration procedure.

The MPU6050 is a widely used chip, and comes with vast documentation and user made coding libraries [10]. It consists of a gyroscope and accelerometer, both used to determine angles. It has the capability of fusing the angle data onboard by using the DMP. Comparisons have been made between the DMP method of data fusion and a complementary filter, with the DMP outperforming the complementary filter [11].

2.3 Useful software

Simulink is a tool often used to simulate control systems. Simscape Multibody, an environment within Simulink, is a useful tool for simulating and visualising the mechanics of rigid multibody systems. Control of mechanical multibody systems is then simulated using a combination of these two tools.

Livescript is useful for computing symbolic equations and expressions. It can be used to determine the equations of motion of a mechanical system, or even model the dynamics of a motor. These equations are then easy to rearrange within the environment and export for use in simulations.

Autodesk have developed Fusion 360: a computer aided drawing (CAD) software which provides an intuitive approach to 3D design. Parts designed, can be exported as STL files ready to be converted into gcode necessary for 3D printing.

3 Mechanical design and hardware assembly

3.1 Overview

Q-TIP needed to be constructed before any testing could be carried out. This required ordering the remaining parts, and machining the cart for attachment. Computer aided design (CAD) software was used to model and 3D print the sensor mounts to be attached without the need for additional machining. A castor wheel was also attached to the base to allow for rotation. A power supply for the system had been set up and placed in the cart. All sensors had been mounted to the top of the pendulum.

An overview of the assembled system is displayed in Figure 2; it captures the mechanical structure and hardware. Table 1 lists all the major components which had to be sourced for the systems hardware and mechanical structure.

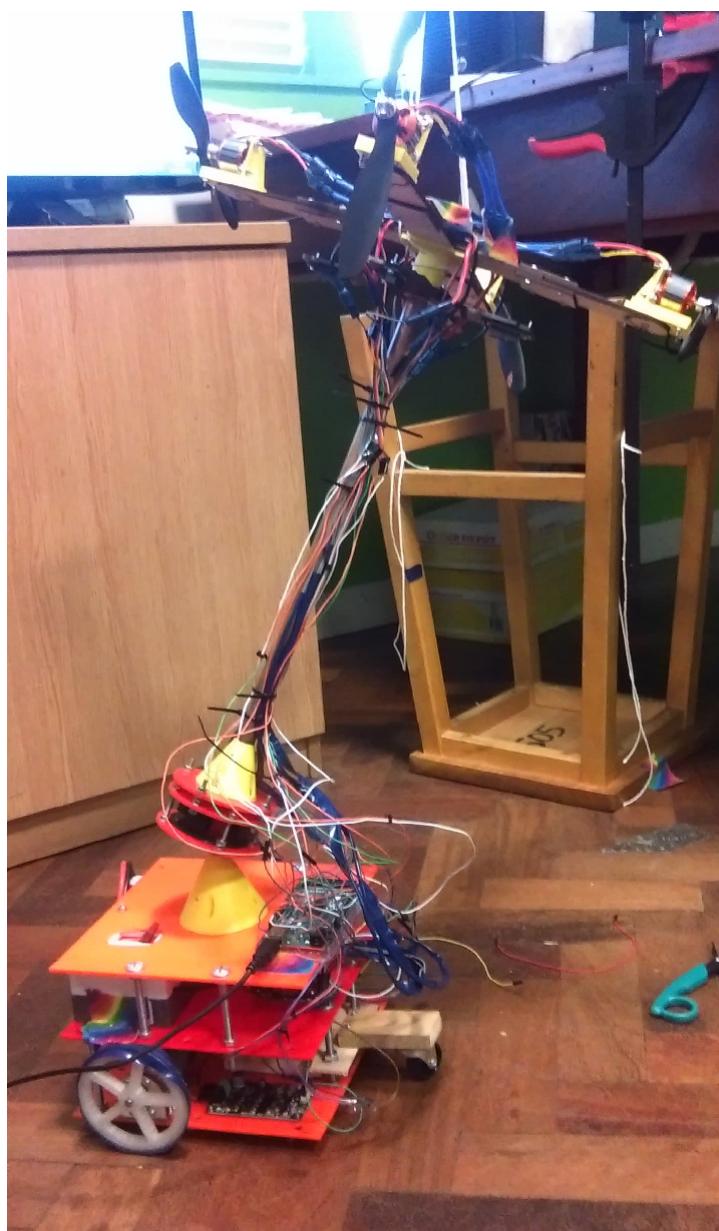


Figure 2: Hardware overview

Table 1: List of major sourced components.

Component	Location	Quantity
RB-Dfr-444 DC motor with encoder	Cart (1st shelf)	2
RB-Dfr-444 DC motor mount	Cart (1st shelf)	2
MD10C 10 DC motor driver	Cart (1st shelf)	2
HC-05 Bluetooth module	Cart (1st shelf)	1
31mm Light duty castor wheel	Cart (1st shelf)	1
Universal 12-way 120A multirotor power distribution hub	Cart (2nd shelf)	1
11.1V 3S 30C 3000mAh LiPo battery	Cart (2nd shelf)	1
80A fused double pole main switch	Cart (2nd shelf)	1
UAV brushless motor A2208/12 1800Kv	Quadrotor	4
8"x4.5" propeller pair	Quadrotor	2
M20A 20A multicopter ESC	Quadrotor	4
MPU 6050 accelerometer and gyroscope	Quadrotor	1
GP2Y0A60SZ infrared distance sensor	Quadrotor	4
GP2Y0A60 carrier board	Quadrotor	4
Arduino Mega 2560	Cart (top)	1

3.2 Cart and wheels

A 31mm light duty castor wheel was attached to the back of the cart. This, in combination with the two drive wheels, balances the cart and allows it to rotate and traverse freely. The castor wheel was attached by a wooden frame which is adjustable in height to match that of the drive wheels (Figure 3).



Figure 3: Castor wheel on wooden frame attachment.

3.3 Power supply

The power supply was mounted on the second shelf of the cart; it consists of three components: the battery, the switch, and the power distribution board (PDB). The battery used was an 11.1V, 3 series cell, 3000mAh, 30C lithium polymer (LiPo), capable of outputting 90A of continuous current. The switch used was fused at 80A since the current drawn from the system should not be able to exceed this. The power distribution board was a 12-way 120A distribution board with additional 5v and 10-20v header pins regulated by universal battery elimination circuits (UBEC).

The switch was connected in between the LiPo battery and the PDB to toggle the system power supply on and off. The switch had to be machined to allow 12AWG wires to pass through each

side; the outlet wires soldered to the PDB, and the inlet wires connected to the battery by a deans connector.

A hole had to be cut into the top layer of acrylic on the cart; this provides access for the switch lever so that it can easily be toggled. The switch was fixed to the top layer of acrylic by means of an M3 bolt, this prevents slip since a significant force is required to flip the switch.

The power distribution board required a good deal of soldering. As previously mentioned, the PDB input voltage (V_{in}) and ground (GND) were first soldered to the switch outlet wires. The UBECs had to be soldered onto the board, along with the 4 header pins for the 5v output, the 4 header pins for the 10-20v adjustable output, and the 8 ground pins. Finally 4 unregulated output voltage (V_{out}) cables, and 4 unregulated GND cables were soldered onto the board to supply power to the four ESCs. The ESC power cables used were the 1.3mm Teflon cables mentioned in [1], with 2.5mm bullet connectors attached at the ends.

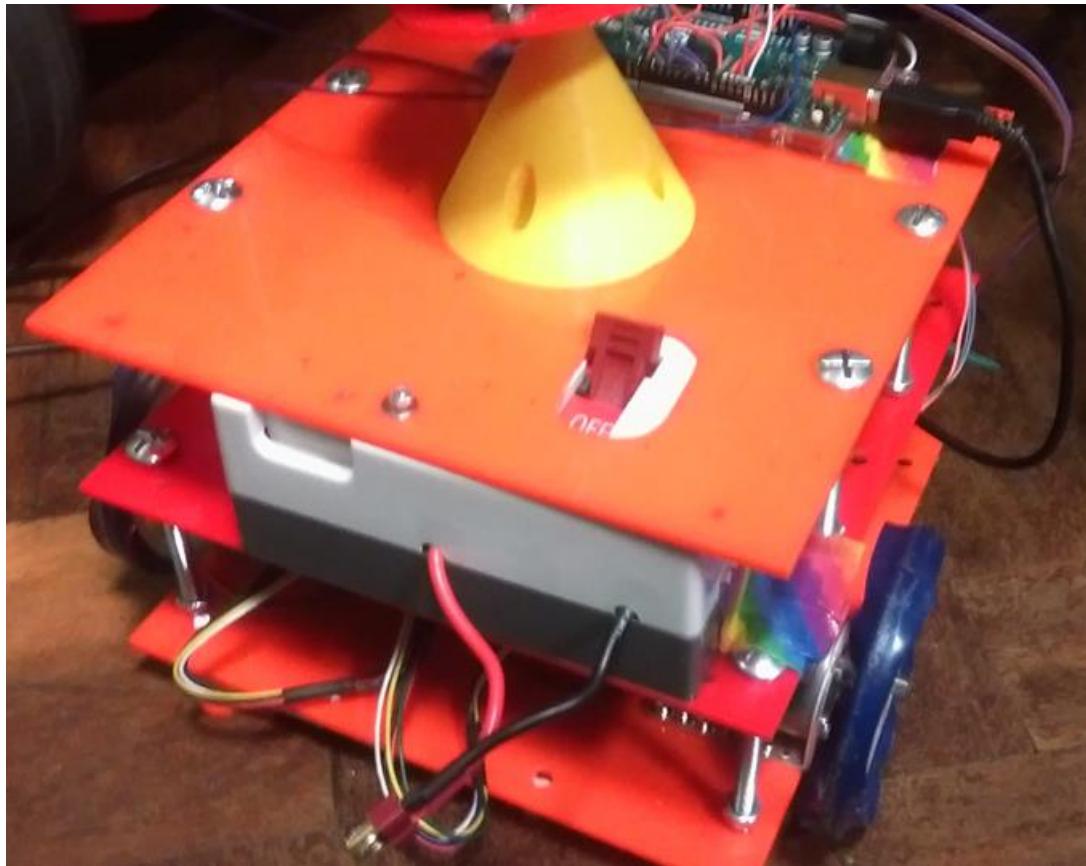


Figure 4: Cart and power supply

3.4 Quadrotor

Originally the quadrotor arms were laser cut from 4mm plywood as in Figure 5. This design fractured through very few uses, therefore a sturdier design was implemented which replaced the slits with holes (Figure 6).

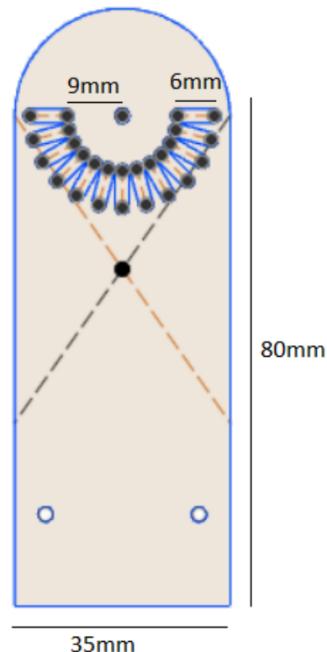


Figure 5: Previous quadrotor arm design. [1]

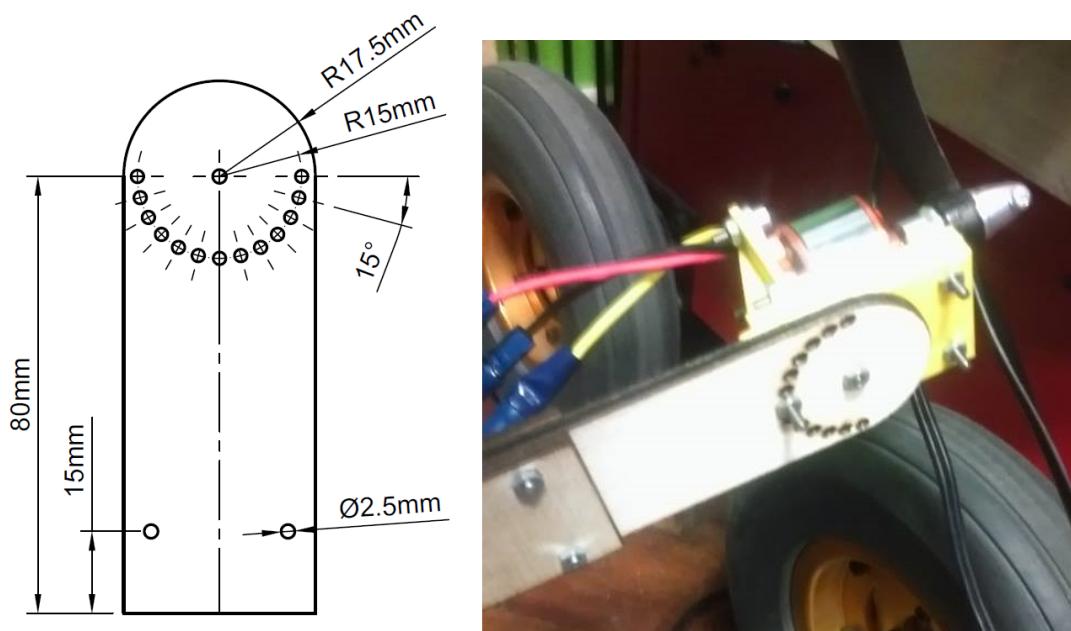


Figure 6: New quadrotor arm design.

3.5 Sensors

Sensing was performed using an MPU6050 gyroscope and accelerometer, along with an infrared (IR) reflective distance sensor. The reflective distance sensor required a reflective plate to be able to measure the angle and distance. The MPU6050 is most suited to be placed in the centre of the quadrotor, so that the coordinate system of the MPU6050 coincides with that of the pendulum arm.

Three components were designed for 3D printing with polylactic acid (PLA): a mount for the MPU6050, a mount for the infrared (IR) reflective distance sensors, and a base cone to mount the reflective plate.

The MPU6050 mount was designed to be attached to the centre of the quadrotor at the top of the pendulum (Figure 7). It accepts the 3mm bolts already used to be attach to the quadrotor, and so no additional machining was required for the attachment. The MPU6050 was then attached to the mount by up to 4 M2 bolts, which are countersunk by 1.5mm so that enough thread is accessible from the other side. A large hole was positioned in the centre of the mount to provide clearance for the protruding parts on the underside of the MPU6050.

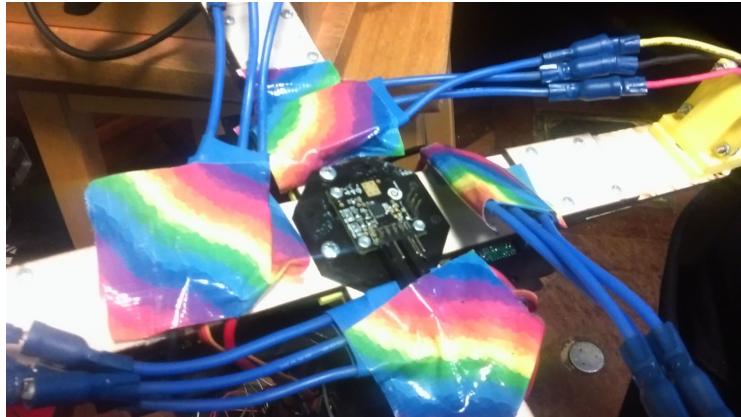
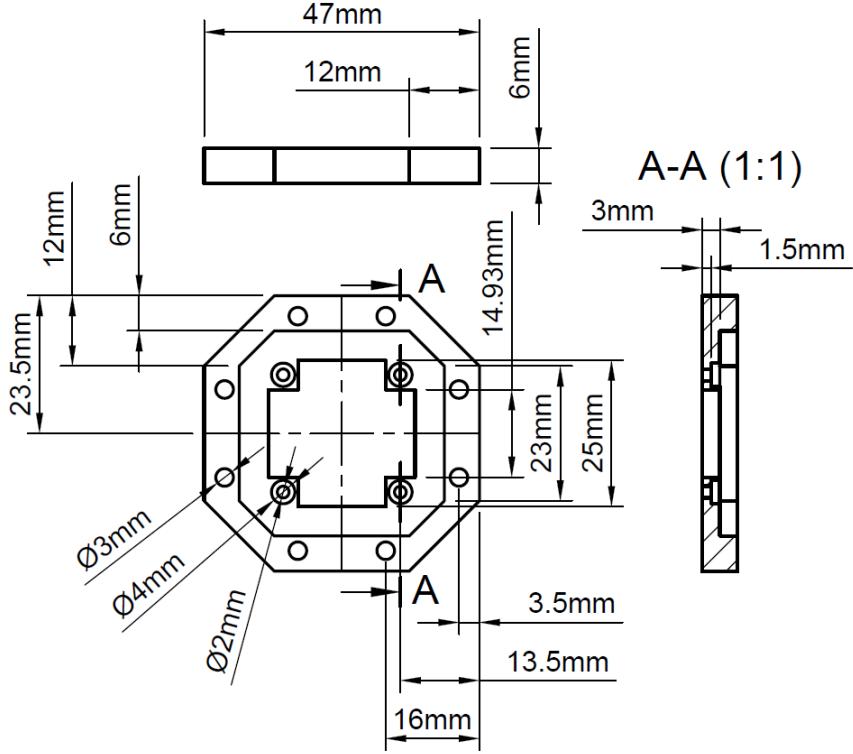


Figure 7: MPU6050 base design and attachment.

A singular part was designed to attach the IR sensors (Figure 8). The part was designed to simply slot over the quadrotor base attachment at the top of the pendulum arm. The sensors work best at greater distances from the surface of reflection and so keeping them at the top of the arm was optimal [12].

Each arm of the IR sensor mount provides an attachment for the IR sensor and breakout board. The IR sensor can be attached in different orientations depending on which performs best. The entire mount is also able to rotate in case other configurations need to be tested.

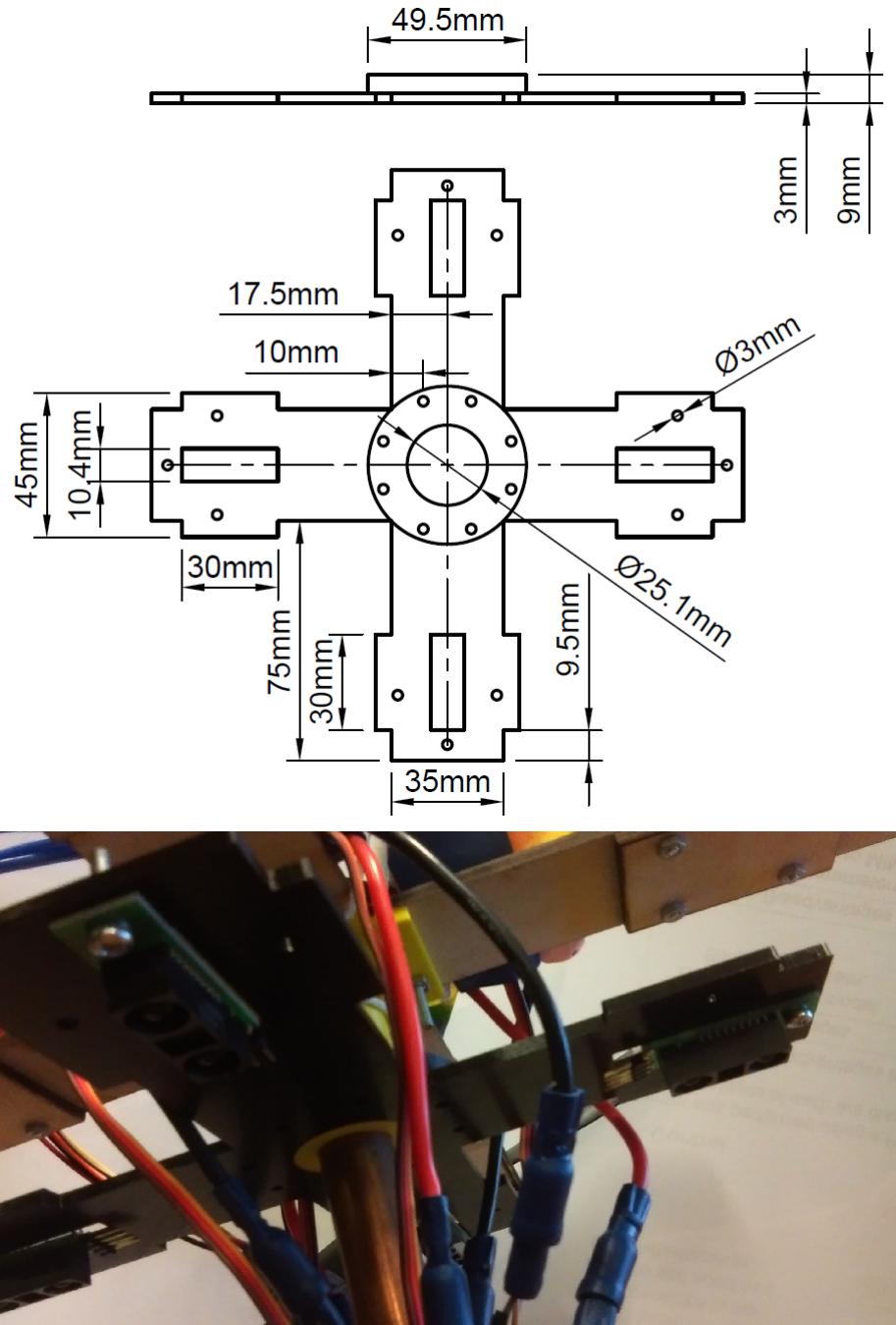


Figure 8: IR distance sensor mount.

The reflective plate from [1] was resized and laser cut from 4mm acrylic, this was because the original design was interfering with the ball joint. Orange acrylic had to be used since the IR sensors cannot reflect off black material. The new dimensions were selected so that the IR distance sensors coincide with the circular centre line, 72mm from the pendulums central axis (Figure 9).

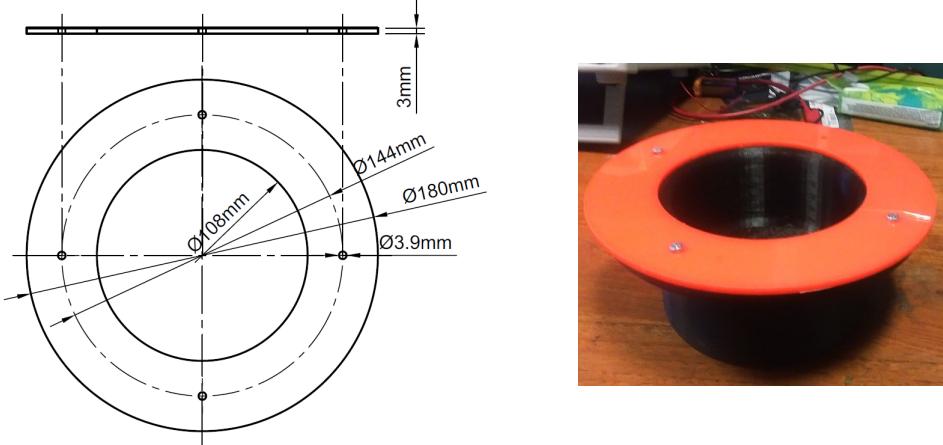


Figure 9: Reflective plate for IR distance sensor.

The reflective plate was mounted on a cone at the base of the pendulum arm. As mentioned in [1], this is to limit the pitch and roll of the pendulum arm by obstructing the ball joint.

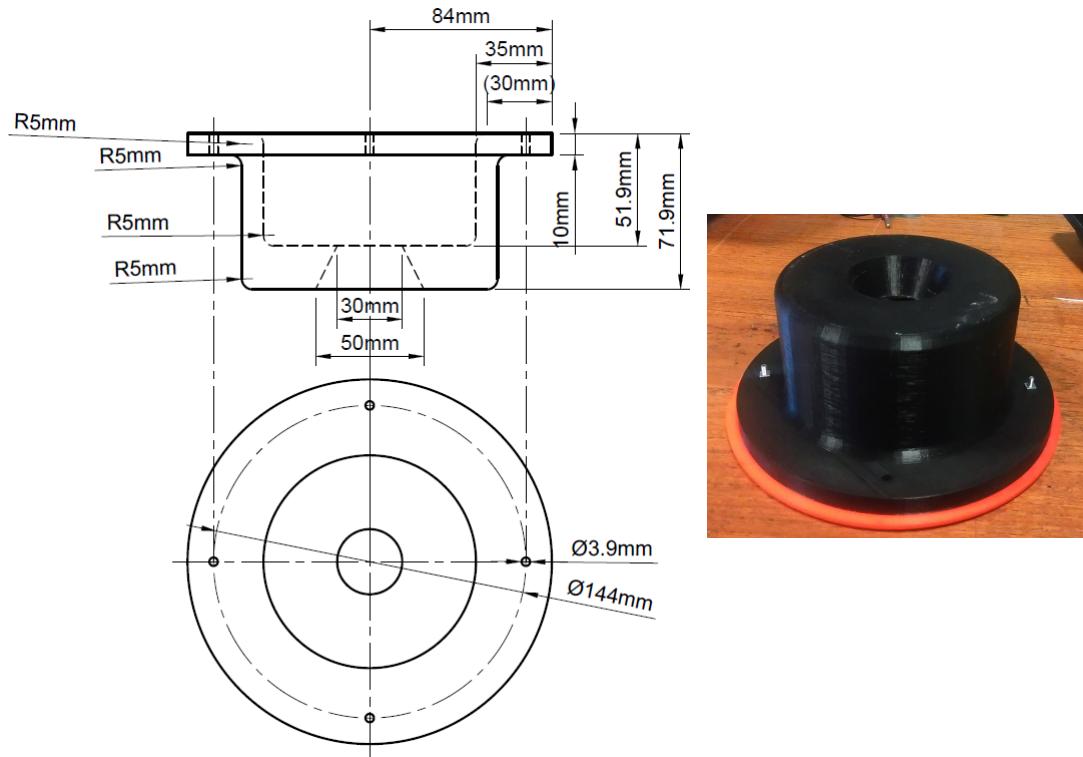


Figure 10: Base cone mount for reflective plate.

3.6 Safety

A risk assessment was completed to implement safety controls and monitor effectiveness of said controls. Safety procedures to adhere to during use of the system are listed below.

The battery output current is high enough to cause fatal injuries to a person or serious damage to an object. To mitigate risk, the battery has to be connected through an 80A fused switch. The switch is connected to a PDB encased inside the cart body. Insulative tape was used on the unregulated power cables to reduce exposure to conductive surfaces.

Battery charging is to take place on a fireproof surface, away from any flammables. It is to be monitored at all times, using the balance connection at 3A with a portable appliance tested (PAT) LiPo charger. Individual battery cell voltage is not to drop below 3.3V. If an individual cell voltage drops below 3.3V then the charger will be unable to detect the cell. In this case the battery is to be charged through the deans connector at very low current, set for 2 cells. Do not let the individual voltage on any cell exceed 4.25V. If any bulging occurs, then the battery is at risk of combustion, and department security must be contacted immediately to isolate it. If combustion occurs, sound the alarm, contact security, and evacuate the building. If an extinguisher is used, be sure it is CO₂.

Testing is performed in a clear area, free from obstructions. Propellers on the quadrotor are fastened tightly, using two sets of pliers. When testing, the quadrotor is set on a timer which cuts pulse-width-modulation (PWM) to the ESCs after a user-set period of time, switching the motors off. The quadrotor is held by string, which acts as a leash, constraining large rotations. In the event of erratic behaviour the power is to be cut off using the main power switch.

Cable ties are to be used to group cables together and keep the wiring out of the way. This prevents shorts and tripping hazards. Light emitting diodes (LEDs) on the PDB will flicker if there are any power issues, which indicates that the user should shut off the power as soon as possible.

4 Subsystems and software

4.1 Overview

A detailed explanation of each subsystem and the associated software is given in this section.

An Arduino Mega 2560 was wired up to all the peripherals using header cables for small power supplies and data exchange. Code is uploaded to the Mega via the Arduino interactive development environment (IDE). The Mega interfaces with the ESCs, ground motors, communicates with the Bluetooth module, and receives data from the sensors at a 115200 baud rate. A mobile app is used as a human user interface with Q-TIP, allowing the user to send commands to the Mega board for cart control. The Mega can alternatively be set up with a PC for command input.

The Arduino code is split into the following sections:

- Inclusion of libraries.
- Definition of variables.
- Setup of MPU6050, ESCs, controller, cart, and starting the internal clock.
- Driving the cart through incoming Bluetooth commands.
- Processing MPU6050 sensor data and filtering.
- PID feedback control.
- Setting ESCs motor speeds.
- Printing to the serial monitor.
- Turning off motors.

Subsystem interaction is illustrated in Figure 11. It shows the relationships of the subsystems and the general flow of the overall system.

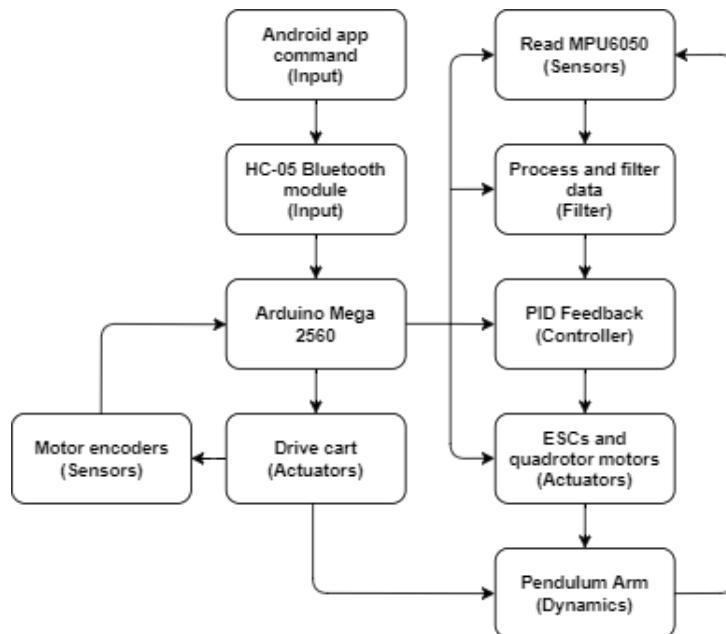


Figure 11: Generalised subsystem block diagram.

4.2 Ground motors

Two ground motors are based in the cart; these have the drive wheels attached to move the cart through differential drive. Each motor has its power terminals connected to a motor driver as displayed in Figure 12. Each driver is powered by a 10-20V UBEC pin on the PDB, set at 10V.

A driver receives a direction signal and pulse-width modulation from the Mega board. The direction pin receives a HIGH or a LOW to indicate which direction the motor is to spin. The PWM pin receives a duty cycle which determines the power the driver supplies to the motor, and hence the speed.

Each motor has an integrated hall effect encoder, this is powered by the Mega board. Connecting the sensor as in Figure 12 allows the hall sensor to send a count signal 2797 times per revolution, [3] and hence the distance travelled by a wheel is given by $s = 2\pi r/2797$ where r is the radius of the wheel. Note that the HALL SENSOR A Vout is connected to pin 2, which on the Mega is interrupt pin 0 (INT.0) [3]. This means that when the encoder needs to send a signal, it can interrupt the Mega so that an encoder reading is not missed, and thus remains accurate.

For the second motor the Mega connections are simply changed from:

$$2 \rightarrow 3, \quad 4 \rightarrow 5, \quad 23 \rightarrow 22, \quad 8 \rightarrow 9.$$

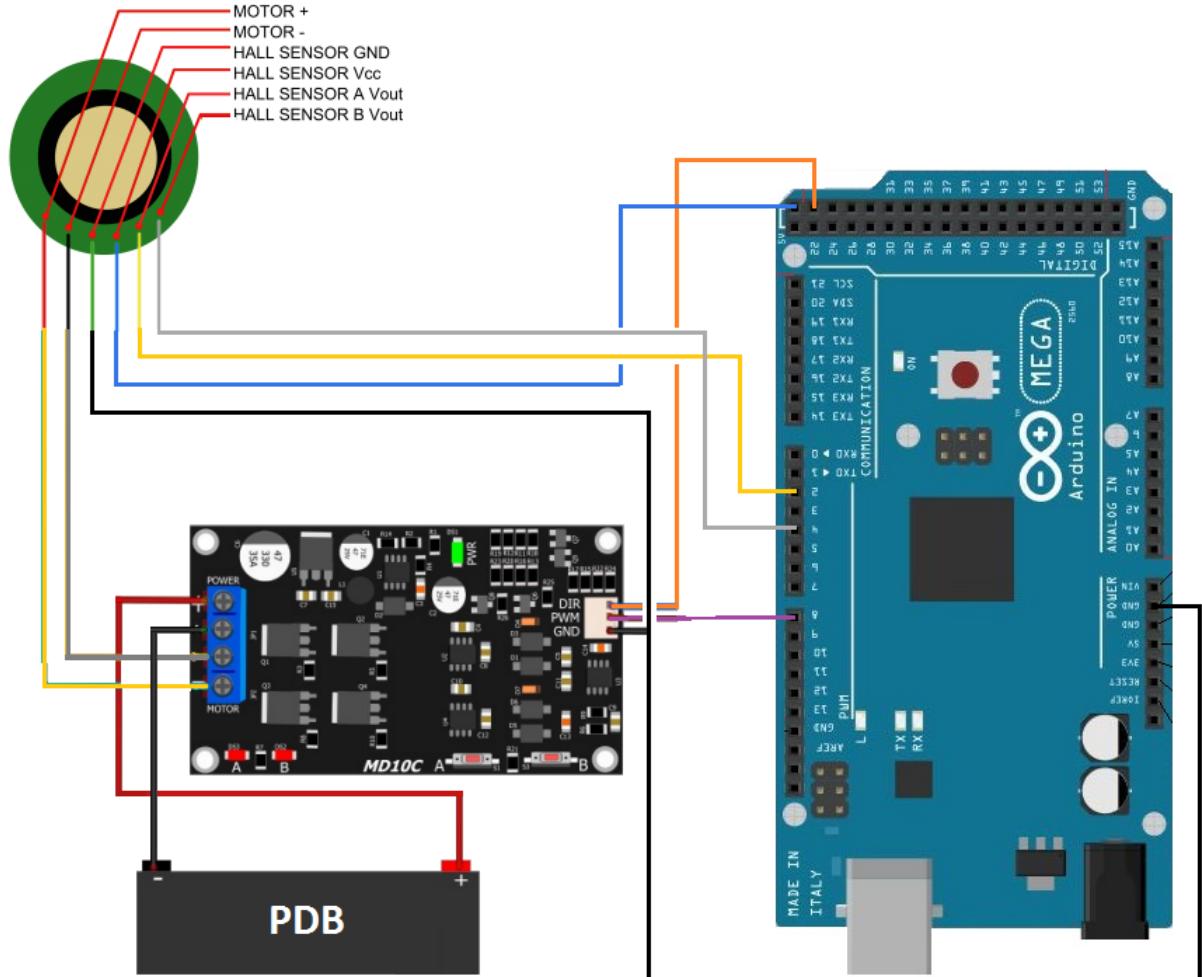


Figure 12: Ground motors and driver setup. (Adapted from [2] and [3])

The Mega board undergoes serial communication with the HC-05 Bluetooth module. The board receives bytes sent to the HC-05 with values from 0 to 9; each value representing a direction in

which the cart is to accelerate. Figure 13 maps out the byte values to directions, with 5 signalling to brake and 0 signalling to decelerate. Each time a byte is received, the speed of each wheel is incremented or decremented according to the direction. In the event that connection is lost, i.e. if no signal is received for 0.5 seconds, the cart will stop completely.

This setup applies accelerations to the cart instead of velocities, so to prevent jittery cart movements which would affect the MPU6050 readings. Increment size, and thus acceleration, can easily be varied in the Arduino code, along with the top speed of the motors.

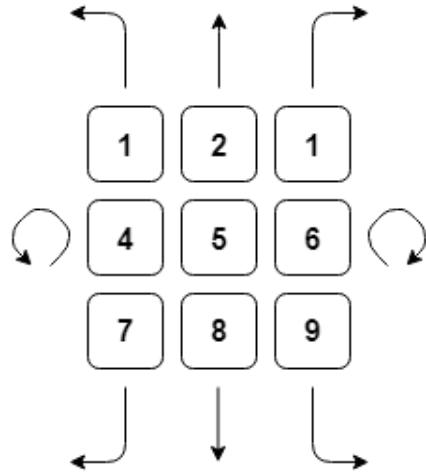


Figure 13: Byte value to direction map.

4.3 Wireless control and interface

User input is sent to the system wirelessly via Bluetooth. An Android app designed in MIT App Inventor 2 is used to send bytes over Bluetooth to a HC-05 model connected to the Mega board.

The HC-05 is configured to act as a slave; it waits to receive a connection from a master, i.e. the Android. Originally the HC-05 initiated serial communications with the Mega board at a 9600 baud rate and so was a bottleneck in the system. Baud rate was increased to 115200 through the AT command mode of the HC-05 [13].

Figure 14 displays the hardware setup of the HC-05 bluetooth module. Note that the cables go from transmitter (TX) to receiver (RX) pins of each device. Pin RX on the HC-05 requires 3.3V, so resistors were added to divide the voltage from 5V to 3.3V and prevent damage. The key pin is used if the user wants to enter the AT command mode to configure the device.

A mobile app graphical user interface (GUI) was developed for Android using the MIT App Inventor 2 (Figure 15). The app sends a signal to the HC-05 every 300ms, in the form of a byte with value ranging from 0 to 9. These byte values are then used to drive the cart. If the 300ms clock triggers with no button held down a 0 is sent.

The app requires the Android to be paired with the HC-05 Bluetooth module, which can be done through the ‘connect’ button. When connected, the app displays the connection status and a counter starts while the connection is active. A signal is sent from the Android every 300ms, which is displayed on screen; this interval can be varied by increments and decrements of 10ms using the GUI.

The HC-05 would often lose its Bluetooth connection with the Android at a 200ms transfer period. This resulted in a broken pipe causing the app to crash. The reason for this could be power issues with the HC-05 since the app still attempts to send a bytes, but the HC-05 does not receive them, yet will sometimes receive them after a delay.

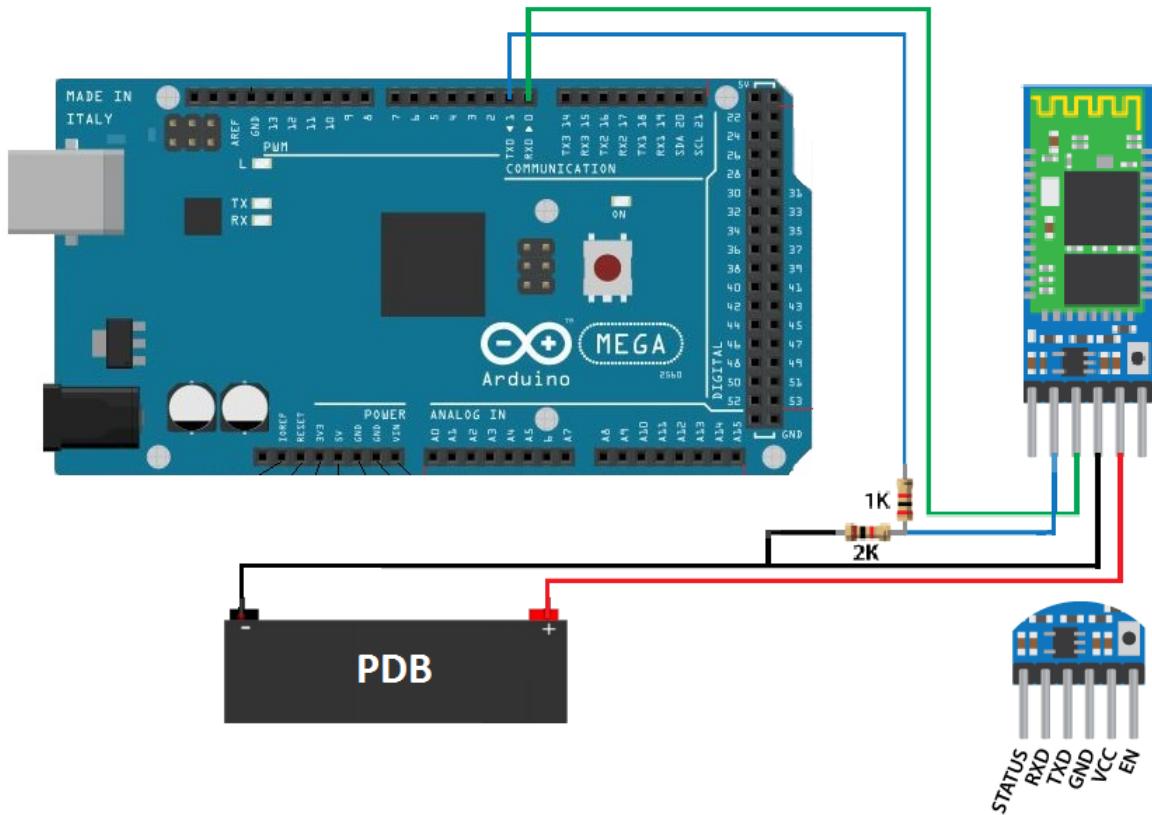


Figure 14: HC-05 Bluetooth connection scheme. (Adapted from [4] and [5])



Figure 15: Android mobile app GUI.

4.4 Sensor readings

Four IR distance sensors are to be used to measure the angle of the pendulum with respect to the cart. Each sensor sends infrared waves from the quadrotor, down to the base of the cart where they are reflected off an acrylic ring. Waves are emitted in all directions, and the amount of energy reflected back to the sensor receiver causes the sensor to produce a voltage. The voltage produced is an indicator of the angle or distance of the sensor relative to the plate.

Sensor voltages can be fed into the analog input of the Mega board as in Figure 16. Each sensor is to be calibrated by taking the mean of a set of voltage readings at different pitch and roll angles. A curve, of an exponential form [14], is fitted to match the data and relate the voltage to angle. In the current wiring solution, the wires obstruct the transmission from the distance sensors to the plate. Readings from the IR sensors are therefore unreliable, and are not used in the feedback.

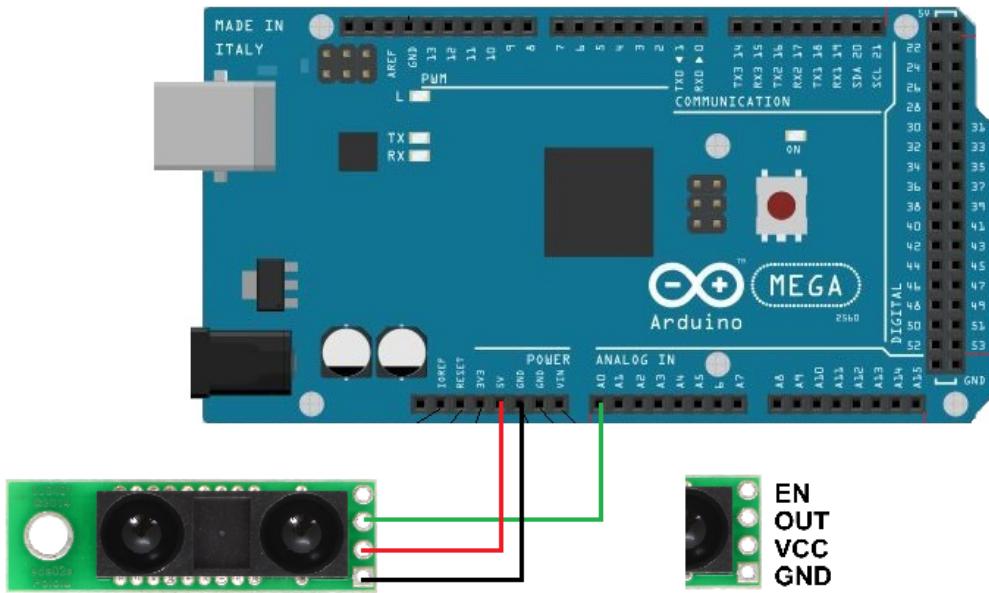


Figure 16: IR distance sensor scheme. (Adapted from [6])

Figure 17 displays the setup of the MPU6050 chip with the Mega board. It is important to note that this setup draws power directly from the board. By doing this, testing of the MPU6050 can be performed without turning on the main power supply for the motors.

The MPU6050 acts as a slave to the Mega board, communicating through an I²C protocol. The MPU6050 sends serial data (SDA) from the SDA pin on the chip, to a SDA pin on the Mega board. The chip synchronises clocks (SCL) with the board through a SCL to SCL connection so that the devices operate at the same rate. A first input first output (FIFO) buffer is used on the chip. When data is generated, it is stored in the FIFO buffer and the interrupt (INT) pin on the chip signals interrupt pin 4 (INT.4) on the Mega board (digital pin 19). The Mega checks to see if the interrupt pin is HIGH, and if so, reads the data from the buffer.

There are two ways to obtain sensor data from the MPU6050. The first way is to read the gyroscope and accelerometer raw data directly from the registers, this requires processing and filtering to be performed on the Mega board. The second way is to use the DMP firmware onboard the MPU6050 chip for direct angle readings.

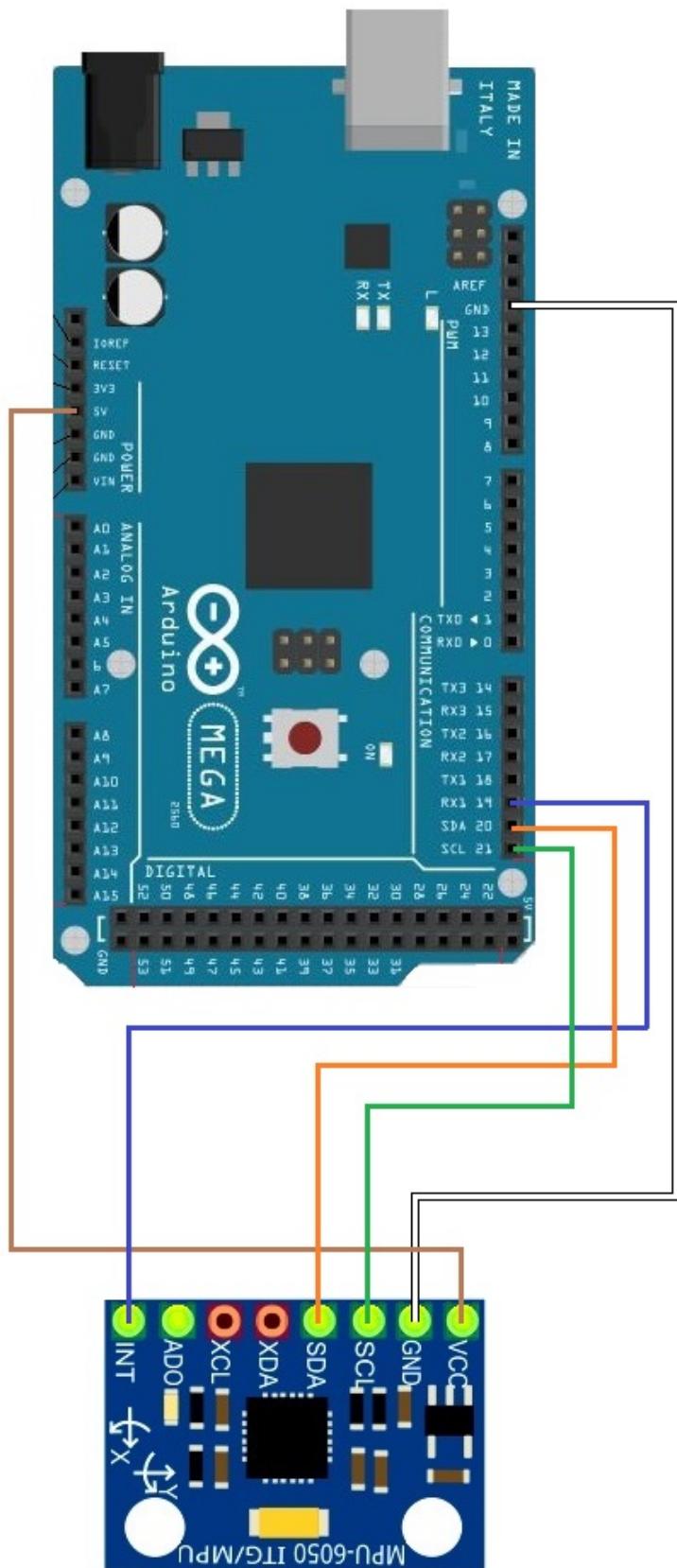


Figure 17: MPU6050 schematic. (Adapted from [7])

4.5 Filtering

Since there are two ways of obtaining sensor data from the MPU6050, there are two different methods of filtering.

Method one: Gyroscope and accelerometer sensor data are read directly from each register of the MPU6050 chip; the data is very noisy and requires some form of filtering or fusion. The Mega board then processes the data into angle readings, i.e. integrates and transforms each reading, and is fuses it using a simple complementary filter. The complementary filter assigns weights to the angle readings from the accelerometer and gyroscope, then combines the two for a fast and smooth response. The fused angle reading is given by

$$Angle_{fused} = (Weight)(Angle_{gyroscope}) + (1 - Weight)(Angle_{accelerometer}). \quad (1)$$

A test was run to observe the filtered data using weight of 0.98. Filtered and unfiltered angle data is plotted in Figure 18. The filtered angle closely follows the gyroscope data since the gyroscope is assigned most of the weighting. It gives a slower response than the accelerometer, but has the perk of running a lot smoother. The accelerometer on the other hand responds significantly faster to changes in angle, but at expense of high amplitude noise. In the filtered response, the accelerometer lessens the delay in the readings, while the gyroscope smooths the noise.

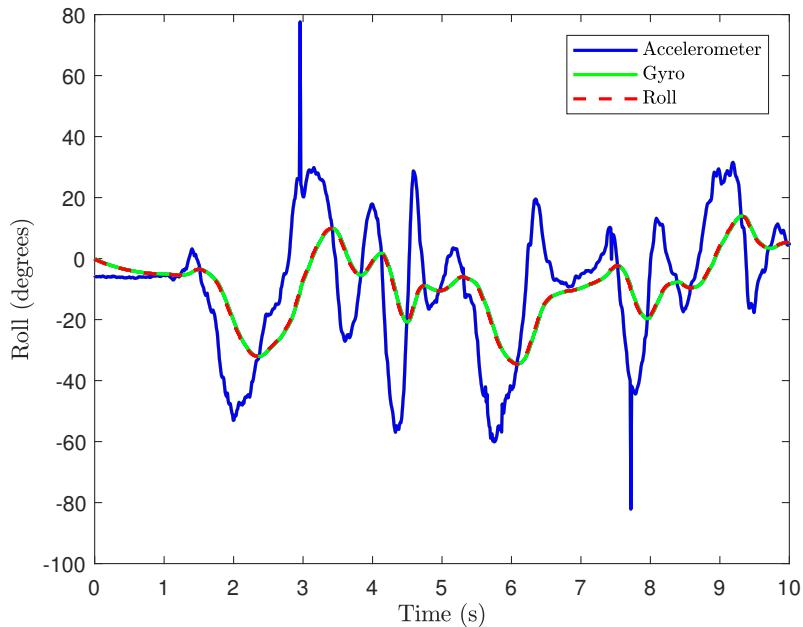


Figure 18: Complementary filter and unfiltered response.

Method two: Sensor fusion takes place directly on the chip, using the DMP. Jeff Rowberg [10] developed a series of libraries which make reading data from the DMP much simpler. His MPU6050 library is used to create an MPU6050 object, making it convenient to read the 42 byte packets from the FIFO buffer. The packets containing data from the DMP can be processed as quaternions or Euler angles; for the purpose of this project Euler angles are sufficient and intuitive. Offsets are also applied to each reading to calibrate the zero angle of the chip. These offsets need to be applied in the setup of every test. If the FIFO buffer were to overflow, a warning is displayed and the buffer is emptied.

Rowerberg's MPU6050 library relies on his 'I2Cdev' library for I²C communication, this in turn relies heavily on the Arduino 'Wire' library. An issue with the 'Wire' library is that if there is a momentary drop in connection, the code will get stuck in a never-ending 'while' loop, known as an Arduino lockup. In the Q-TIP hardware, signalling cables run the entire length of the pendulum

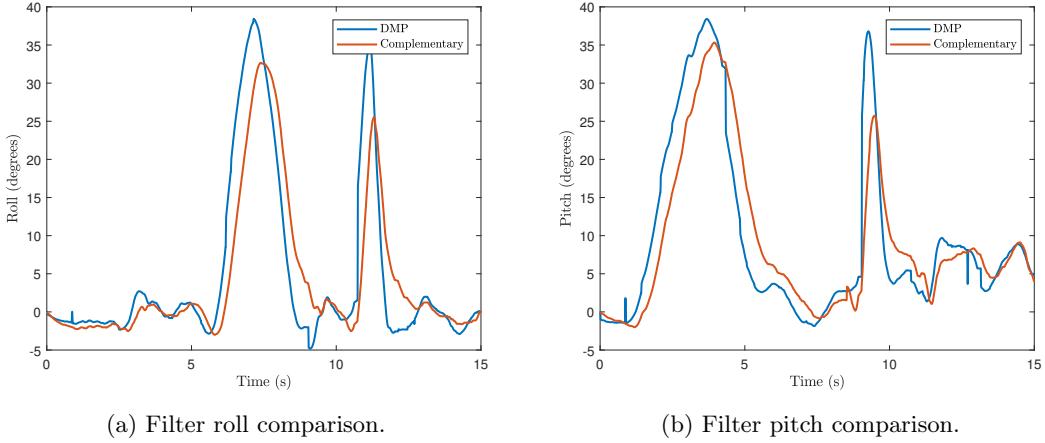


Figure 19: Filtering method comparison.

arm, from the Mega board based on the cart, to the MPU6050 chip on the quadrotor. Due to the cable length, the voltage drop often causes a temporary signal loss, and so the ‘Wire’ library causes the code to lockup. The solution to this was to alter the ‘I2Cdev’ library and replace the dependencies on the ‘Wire’ library with the ‘I2C master library’ by DSSCircuits [15]. This ‘I2C master library’ comes with a configurable timeout, so that if the code hangs for too long the loop will break, and prevent Arduino lockup.

The timeout in the new library caused some bytes to not be read, this resulted in an incomplete packet of less than 42 bytes and caused erratic DMP readings. To fix this, the code on the Mega board empties the buffer if the byte count is not a multiple of 42. If the byte count exceeds 1024 in the buffer, the buffer is full and gets emptied.

Figure 19 displays the roll and pitch time responses of a single 15 second simulation. Comparing the two methods it is found that the DMP fusion is fastest since computation is handled on the chip, its readings are also more robust to cart movements. The complementary filter is slower; it lags behind the DMP, and is strongly affected by the cart movements. The complementary filter can also combine data from the IR distance sensors which holds an advantage over the DMP. Since the IR distance sensors are unreliable in the current hardware setup, the DMP is preferred.

4.6 ESCs and top motors

Motors atop the quadrotor are controlled by 20A ESCs (Figure 20). The motors spin the propellers to provide lateral thrust and are therefore the main power sink. The ESCs need to draw power from the PDB; they draw unregulated power through the bullet connections since the ESCs have their own inbuilt battery eliminator circuits (BECs). Two propellers need to spin clockwise while the other two spin anticlockwise to provide a forward thrust. Swapping the red and yellow wires running from the ESC to the motor reverses the polarity, thus direction of the motor.

A single header cable is attached to the Mega for PWM. The Mega emits pulses at a duty cycle which depends on the PID control feedback law (or whichever control law is used). Another ground connection is attached to a UBEC GND pin on the PDB by header cable. These cables cannot be too long otherwise the voltage drop will cause the ESCs to lose connection and switch off.

Input from the Mega to the ESC has to be calibrated every time the device is powered on. The ESCs accept an integer in the range of 600 to 2000 from the Mega board. 2000 results in the highest speed, and 600 results in the lowest, anything below switches the motors off. Calibration is performed as such: send max speed (2000) integer to the ESC, power on the ESC, clear the serial buffer (or send minimum integer value of 600).

As in Figure 20, the other three ESCs have signal pins connected to PWM pins 11, 12 and 13 on

the Mega board.

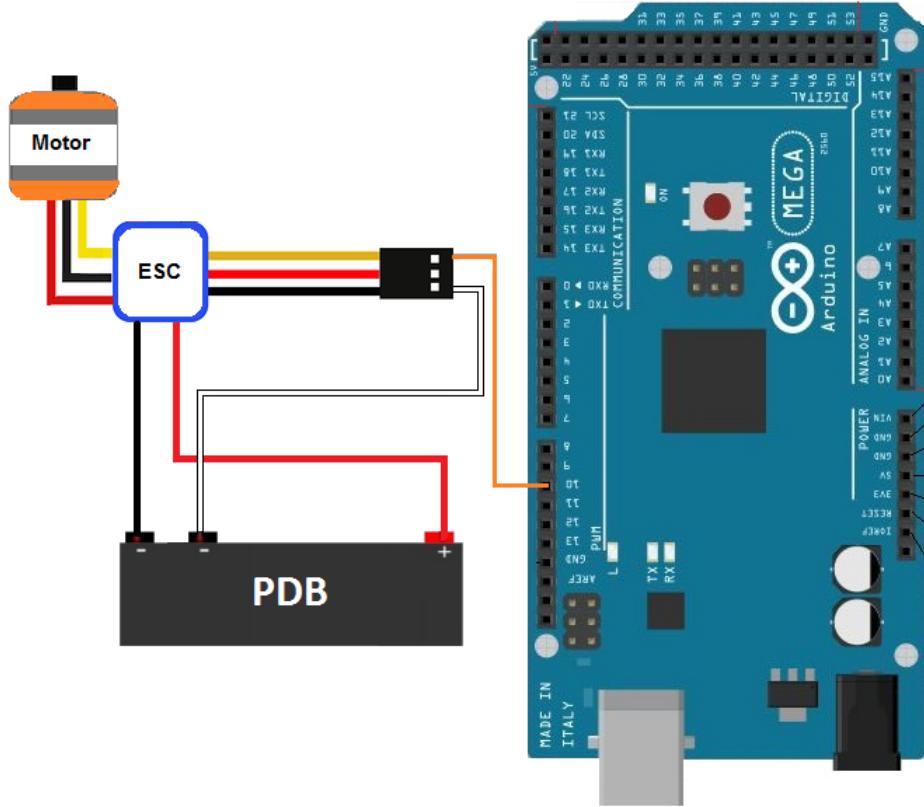


Figure 20: ESC and quadrotor motors scheme. (Adapted from [8])

4.7 Feedback control

Feedback is carried out on the Mega board using a PID control scheme. Angle readings are input into the controller, and PWM signals are output to the ESCs. The user sets the proportional, integral and derivative gains before the code is uploaded to the Mega board. Proportional, integral and derivative gains of 41, 0.0005 and 8000 respectively are sufficient to stabilise the system.

Although the top motors start spinning when the Mega sends integers of 600 or more to the ESCs, negligible thrust is produced until integers of approximately 1000 or more. The ESC inputs are saturated with a lower bound of 1000 and an upper bound of 1800, limiting the difference in thrust of opposite propellers, thus the net force. This limitation gives a smoother response, allowing the sensors to produce more accurate and less delayed readings.

5 Hardware Operation

5.1 Operating procedure

Procedures for operating Q-TIP vary depending on which tests are to be carried out. Each module, e.g. the MPU6050, is enabled by calling its associated function in the Mega operating loop. The general procedure is as follows:

1. Comment or uncomment the function calls to enable or disable modules.
2. Set tunings, ESC saturation range, ground motor speed saturation range and acceleration, end time of test, and pick which outputs to observe.
3. Upload the code to the Mega board and await instruction to turn on the power.
4. Ensure appropriate safety precautions are in place.
5. Position the system in the desired starting position.
6. Turn on the power when prompted by the Arduino IDE's serial monitor. The ESCs should quickly beep three times, pause for a second, and beep a fourth time. Connect the app through Bluetooth if desired.
7. Start the test by sending any signal via serial communication, be it through app or PC. If using the PC, make sure the Mega's RX0 (0 pin) is disconnected.
8. Observe test until timer powers system off. If unstable then immediately turn the power off.

Note that setup of the modules occurs after the code is uploaded to the Mega, but before the power is supplied to the ESCs. This is due to the fact that the ESCs have to be calibrated on every startup. Generally the code loop executes at 100Hz, which matches that of the DMP to FIFO frequency programmed in. It is then most appropriate to set the I²C communication to timeout at 20Hz (50 ms) if no data can be read.

5.2 Troubleshooting and common problems

MPU6050: Sometimes the Arduino loop will slow down. This occurs when there is a high volume of serial traffic, e.g. using 'Serial.print' function repeatedly. It may be worth using one of the Mega's additional serial ports [5].

HC-05: The HC-05 will sometimes not power up, or will stop receiving Bluetooth data (indicated by a "Broken Pipe" error message). It may be worth connecting the HC-05 device to another power source or header pins. It may also be due to the Bluetooth master device running the app, and so alternative devices could be tried. Also note that the PC cannot commence serial communication with the Mega board while the HC-05 is transmitting to Mega pin RX0 (pin 0). The PDB is used to power the HC-05 so that the PC can communicate with the board while the main power is off. It may be worth reconnecting the HC-05 TX and RX pins to different TX and RX pins on the Mega board, such as TX3 and RX3.

PDB: If the LEDs on the PDB begin flickering, turn off the power immediately. The flickering is caused by powering issues. Sometimes the ESCs will not shut off automatically at low battery power, and so the most common source of LED flickering is low battery. Although the following should not happen; it is important to make sure nothing has come loose, and that cables are not overlapping.

ESCs: If a signal cable is too long, the ESC will not receive the signal and so will emit a single beep and not spin after powering on. If the signal ground cable is too long, the ESCs will emit

a single beep after powering on. Once a speed is sent by the controller, the ESC will beep at high frequency and begin to shake. Connections of the signal cables are loose and can sometimes disconnect.

Ball joint: After much wear, the ball joint is prone to detaching from the cart during operation, and so it needs to be replaced.

Nuts and bolts: These are all prone to loosening after repeated use and need to be tightened every so often.

6 Modelling and simulations

6.1 Modelling

Models were derived to capture the dynamics of the mechanical part of the system. Equations of motion were first found using MATLAB Livescript. These were used to build a Simulink model and simulations were verified by comparing them to a Simscape Multibody model.

Assumptions had to be made for the modelling, they are as follows:

1. Pendulum geometry and kinematics are assumed symmetric through the X-Z and Y-Z planes.
2. The centre of gravity and rotation of the cart coincides with the centre of the ball joint.
3. All bodies are assumed rigid.
4. The cart cannot move in its vertical direction or lateral direction (no sideslip).
5. All angles of rotation are unconstrained.
6. Frictionless ball joint and wheels.

To find the equations of motion, first a set of coordinates need to be defined (Figure 21). The system is modelled as two bodies, with one set of coordinates defined to describe the cart, and another for the pendulum arm. Note that the coordinate system on the pendulum matches that of the MPU6050, and so an angle of $\psi = \gamma$ means that the yaw of the pendulum is 0 with respect to the cart (both share the same heading).

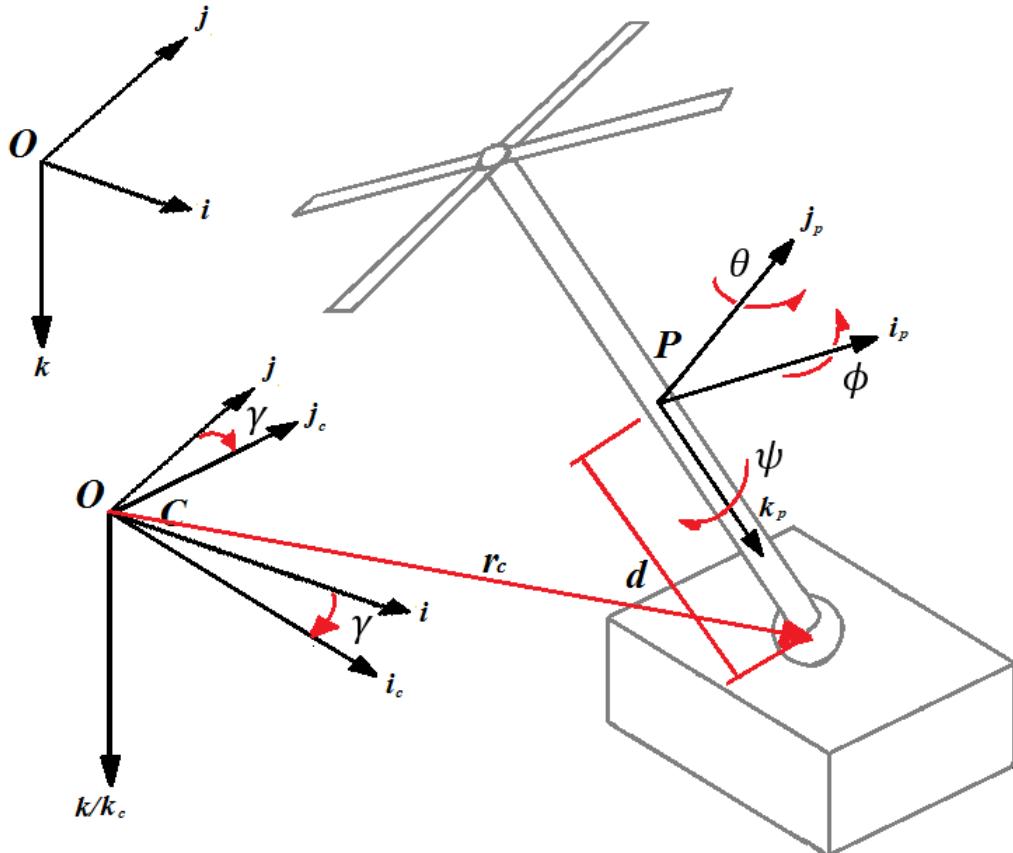


Figure 21: Coordinate systems and transformed frames.

Transformation matrices \mathbf{T}_B , \mathbf{T}_C and \mathbf{T}_D convert angle readings from the Earth-fixed frame \mathbf{O} , to the pendulum body-follower frame \mathbf{P} . The transformations are carried out under the yaw-pitch-roll convention. First \mathbf{T}_D induces a yaw about the z -axis, second \mathbf{T}_C induces a pitch around the intermediate y -axis, and finally \mathbf{T}_B rolls about the new x -axis.

$$\mathbf{T}_B(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}, \quad \mathbf{T}_C(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}, \quad \mathbf{T}_D(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Conversion from the body frame, back to the Earth frame is simply done by inverting the transformation matrix, which is equivalent to taking the transpose, i.e $\mathbf{T}^{-1} = \mathbf{T}^T$. The position vectors with respect to the Earth frame can be found by finding the unit vectors

$$\mathbf{k}_p = (\mathbf{T}_B(\phi)\mathbf{T}_C(\theta)\mathbf{T}_D(\psi))^T \mathbf{k}, \quad (2)$$

and computing the position vectors

$$\mathbf{r}_c = x' \mathbf{i}_c + y' \mathbf{j}_c = \mathbf{T}_D^T(\gamma) [x' \quad y' \quad 0]^T, \quad (3)$$

$$\mathbf{r}_p = \mathbf{r}_c - d \mathbf{k}_p. \quad (4)$$

Taking the time differential of the position vectors with respect to the Earth fixed frame gives the velocity vectors

$$\mathbf{v}_c = \dot{\mathbf{r}}_c. \quad (5)$$

$$\mathbf{v}_p = \dot{\mathbf{r}}_p = \mathbf{v}_c - d \dot{\mathbf{k}}_p. \quad (6)$$

Lagrangian formalism [16] is used to compute the equations of motion: with kinetic energy

$$T = \frac{1}{2} m_c \mathbf{v}_c^T \mathbf{v}_c + \frac{1}{2} m_p \mathbf{v}_p^T \mathbf{v}_p + \frac{1}{2} J_z^c \dot{\gamma}^2 + \frac{1}{2} J_x^p \dot{\phi}^2 + \frac{1}{2} J_y^p \dot{\theta}^2 + \frac{1}{2} J_z^p \dot{\psi}^2, \quad (7)$$

and potential energy

$$V = -m_p g d \cos \phi \cos \theta. \quad (8)$$

The Lagrangian function is then simply

$$L = T - V. \quad (9)$$

Lateral movement of the cart is constrained by the nonholonomic no-slip constraint. It is found by setting the \mathbf{j}_k component of \mathbf{v}_c to zero, and so is defined as

$$C = \dot{y}' + \dot{\gamma} x' = 0. \quad (10)$$

Equations of motion of the system, using the generalised coordinate $q \in \{\phi, \theta, \psi, x', y', \gamma\}$, are given by

$$\frac{d}{dt} \left(\frac{dL}{dq} \right) - \frac{dL}{dq} + \lambda \frac{dC}{dq} = \sum F \quad (11)$$

for translations, and

$$\frac{d}{dt} \left(\frac{dL}{d\dot{q}} \right) - \frac{dL}{d\dot{q}} + \lambda \frac{dC}{d\dot{q}} = \sum \tau \quad (12)$$

for angles. A MATLAB Livescript is used to compute and solve the equations of motion by substituting in the constraint and factoring out λ . The equations are highly nonlinear and are far too large to be included in the report.

6.2 Model verification by simulation: 1 DOF pendulum on 1 DOF cart

By making the substitutions, the equations of motion can be reduced. For the 1 DOF case, with:

$$[\phi \ \dot{\phi} \ \ddot{\phi} \ \psi \ \dot{\psi} \ \ddot{\psi} \ \gamma \ \dot{\gamma} \ \ddot{\gamma} \ y' \ \dot{y'} \ \ddot{y'}] = 0, \quad (13)$$

the equations of motion can be reduced to the 1 DOF case. The equations of motion in 1 DOF are identical to that of an inverted pendulum:

$$(m_c + m_p) \ddot{x}' + m_p d \dot{\theta}^2 \sin \theta - m_p d \cos \theta \ddot{\theta} = (F_l + F_r) - F_\theta \cos \theta \quad (14)$$

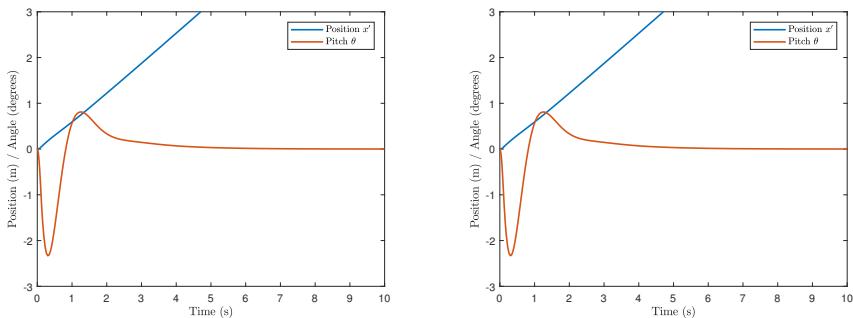
$$(m_p d^2 + J_y^p) \ddot{\theta} - m_p \ddot{x}' d \cos \theta + m_p g d \sin \theta = F_\theta l. \quad (15)$$

Simulations were carried out for the single plane version of the system. A Simulink model was developed from the equations of motion 14 and 15. A Simscape Multibody model was also built to validate the Simulink model. Simulations in the Simscape model provide a 3D graphical simulation of the response. Simulations were ran in both models using the parameters listed in Table 2.

Table 2: Simulation parameters for 1 DOF.

Parameter	Value	Units
m_c	1	kg
m_p	1	kg
J_y^p	1	kg m ²
d	1.5	m
l	3	m
g	9.81	m s ⁻²

Simulations were ran in Simulink and Simscape subject to a step input of $F_l + F_r = 10$ N, beginning at $t = 0$ s and stopping at $t = 0.1$ s. A PID control law was implemented using the angle error in both of the models; this control is fed back into the propeller thrust F_{pitch} . The proportional, integral and derivative gains used were respectively 10, 10, 5 in both models, so that they remained identical. Plots in Figure 22 show the state time responses under a 10 second simulation.



(a) Simulink state response.

(b) Simscape state response.

Figure 22: Simulations of 1 DOF pendulum on 1 DOF cart.

Simulations in Figure 22 show identical, completely stable responses. They match what is expected to be seen; as well as the 3D graphical simulation. A quick acceleration for the first 0.1 s while the force is applied, and then constant velocity for the cart. The pendulum arm is stabilised by the PID controller, which regulates the angle to 0 as intended, regardless of the carts motion. Using the MATLAB function ‘linmod’ results in identical linearised models, further supporting the validity of the models. Overall the single plane pendulum and cart model is sufficiently accurate to be used in the real system.

6.3 Model verification by simulation: 3 DOF pendulum

The pendulum portion of the model also needed to be validated. To obtain this part of the model, the substitution

$$[x' \quad \dot{x}' \quad \ddot{x}' \quad y' \quad \dot{y}' \quad \ddot{y}' \quad \gamma \quad \dot{\gamma} \quad \ddot{\gamma}] = 0 \quad (16)$$

is made. Simulations of the system are performed using the parameters listed in Table 3. First a simulation was performed for 10 seconds, with initial conditions $(\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}) = (\pi - 0.5, 0, \pi - 0.2, 0, 0, 0)$ representing a small deviation from the downward position. No input force was applied and so Figure 23 shows the free response.

Table 3: Simulation parameters for 3 DOF pendulum.

Parameter	Value	Units
m_p	2	kg
J_x^p	0.0001	kg m ²
J_y^p	0.0001	kg m ²
J_z^p	0.1	kg m ²
d	0.35	m
l	0.5	m
g	9.81	m s ⁻²

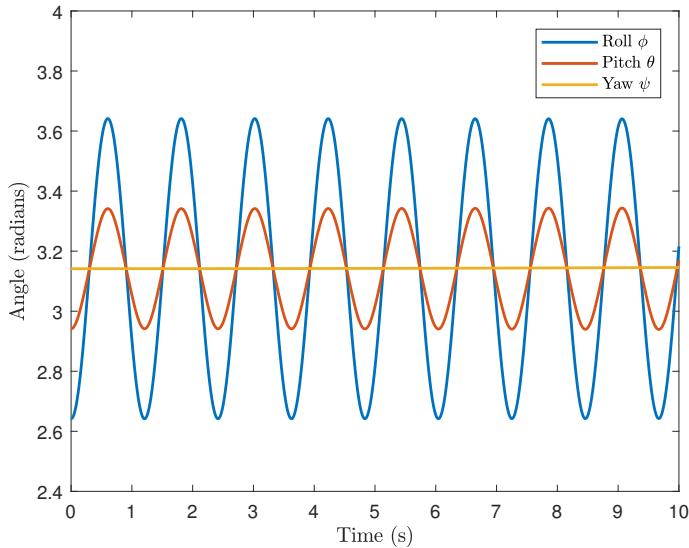


Figure 23: Unforced response with constant yaw.

Yaw remains constant throughout since there is no initial velocity and no force to drive it. Roll and pitch oscillate about π at a constant amplitude, as expected of an unforced and undamped pendulum.

A similar simulation was run (Figure 24), the only difference being an initial yaw velocity: $(\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}) = (\pi - 0.5, 0, \pi - 0.2, 0, 0, 0.5)$.

As yaw increases, the roll and pitch behaviours vary. Oscillations still occur about π . An increasing pitch results in a decreased roll, and vice versa. When the yaw angle reaches a rotation of π , the pendulum direction is reversed about the z axis.

The behaviour appears as it is expected to be seen, but unfortunately Simscape could not be used to verify it. Simscape does not allow the user to set an initial velocity for the yaw and so could not generate valid results. However, it did produce identical results to the Simulink model for a constant yaw.

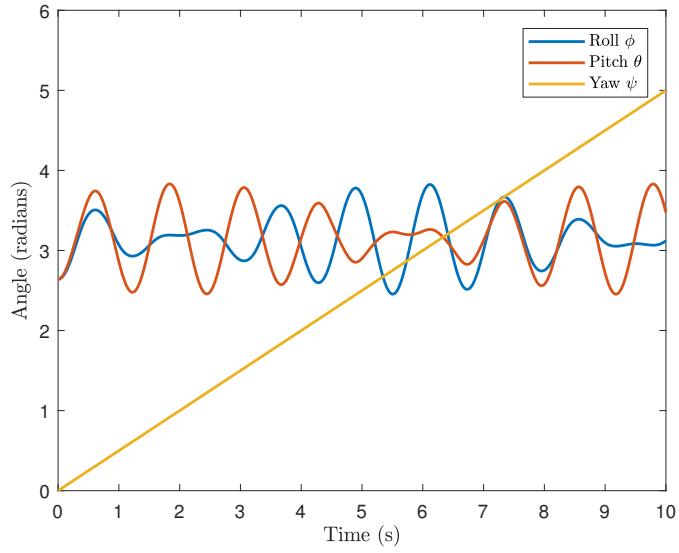


Figure 24: Unforced response with constant yaw rate.

These simulation results combined indicate that the equations of motion are reliable enough to be used for a model based control approach. The model appears sufficiently accurate, and once parameters are estimated, the models can be used to design controllers.

7 Hardware response

Four tests were performed on the hardware with the propellers mounted facing directly outwards. One test for the roll response, one for the pitch response, one for their combined response, and one to test the drift correction. Each test was carried out using the DMP for sensor fusion, and the cart kept stationary. PID feedback was used with proportional, derivative, and integral gains of 41, 0.0005, and 8000. The steady state target for roll and pitch is 0° .

7.1 Roll error feedback performance

Figure 25 plots a 10 second time response starting from a large roll angle of -26° . It demonstrates that the roll angle is capable of stabilising from a large deviation from the equilibrium. It takes the system roughly 4 seconds for the system to reach steady state. The steady state offset of the roll angle is 3° , and offset of the pitch angle is 4° . These offsets are likely due to a combination of calibration error, sensor drift, and starting the DMP from a large angle. The yaw reading also reaches a steady state, but has no exact target, nor does it have any means of control to reach the target.

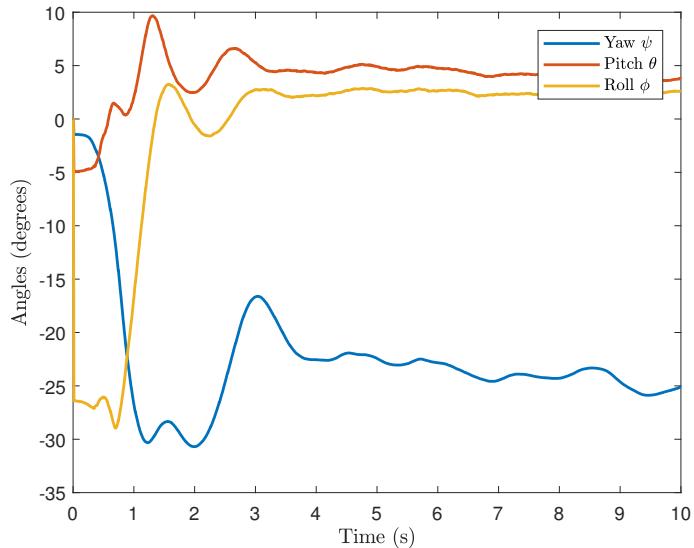


Figure 25: Hardware response from large initial roll.

7.2 Pitch error feedback performance

Figure 26 plots a 10 second time response of the system starting from a large initial pitch of 26° . This time steady state is reached in only 3 seconds. The pitch and yaw are again stabilised with steady state offsets of 3° and 1° . Yaw and roll show very little overshoot and the responses appears relatively smooth; the result being less sensor drift and thus a smaller offset.

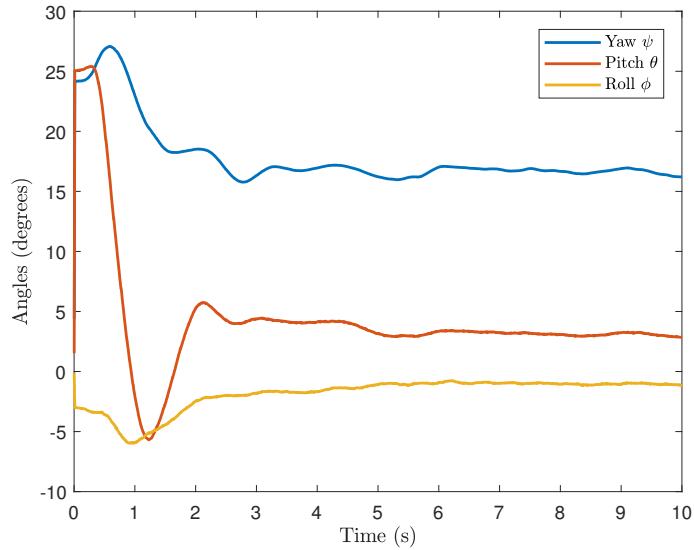


Figure 26: Hardware response from large initial pitch.

7.3 Combined roll and pitch error feedback performance

A combination of the roll and pitch angle responses are tested in Figure 27, for a runtime of 10 seconds. This starts the pendulum arm with roll and pitch angles of 17° and -14° respectively. Roll and pitch steady states are reached rather quickly at 2.5 seconds. Yaw shows a decreasing response as the yaw rate approaches a constant negative value. Lack of control in the yaw direction means that friction from the ball joint is the only way for yaw to reach steady state. Steady state offsets produced for roll and pitch are 2° and 4° respectively.

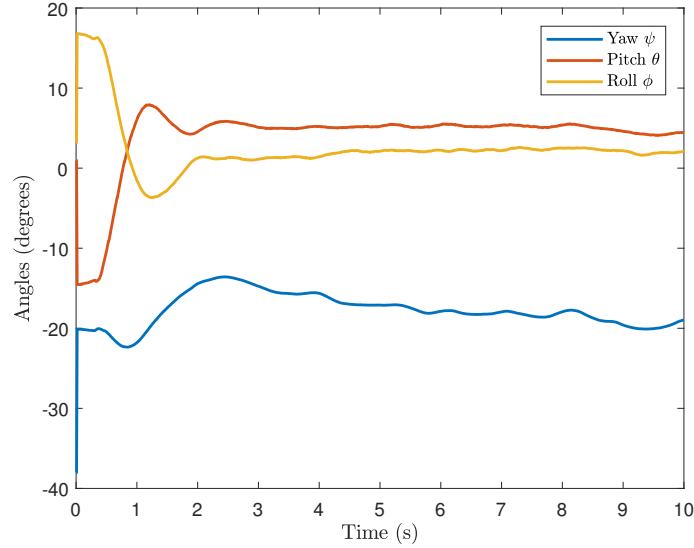


Figure 27: Hardware response from large initial roll and pitch.

7.4 Steady state tracking performance

A test was also run to observe sensor drift and steady state tracking error over time (Figure 27). This time the test ran for 30 seconds, starting from as close to the equilibrium as humanly possible. The pitch and roll angles quickly drifted from the equilibrium initially, likely due to the impulse of the user letting go over the pendulum arm. The feedback quickly tries to counter the initial angle deviation, with the small integral gain playing a bigger part in tracking. It can be seen that over the entire 30 second period, the integral feedback decreases the tracking error, counteracting drift. Disturbances and noise are a lot more observable in this plot, due to the relatively small angles. Increasing the integral gain decreases the steady state tracking error, but at the cost of worse performance when subject to large pitch and roll angles.

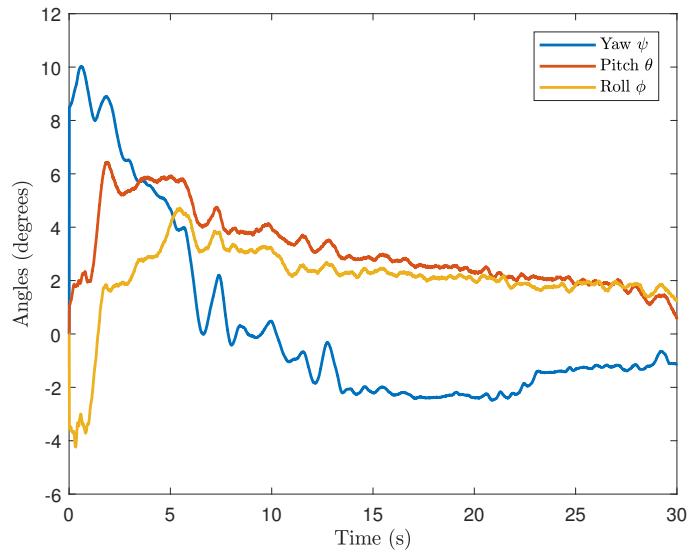


Figure 28: Hardware response from equilibrium initial condition.

8 Conclusions

8.1 Summary of Q-TIP

Overall the aims of the project were met. The system was successfully constructed and stabilised with PID control. A method of driving the cart with a GUI was also developed as intended. The system hardware remains modular, with parts easy to replace with newer or alternative variations. The parts used are commonly used parts with a large availability of documentation online.

There are a wide range of online resources such as libraries and code free to access, for example the MPU6050 libraries. These libraries and code do require modification to fit the user need, as they are still in development.

Modelling resulted in large equations of motion which can be reduced depending on how many DOF the system is configured for. These are ready to use for model based controller design such as LQR or H_∞ . Although the equations of motion are highly nonlinear, and a linear controller would need to be rather robust, and so LQR is unlikely to perform well. Linear controllers may be hard to implement for stabilising the pendulum arm around a deviation from the upright position.

PID control proved to be an effective method at stabilising the pendulum arm at a constant reference, but would need to be tuned for the new reference. It would also need to be tuned whenever a parameter were to be significantly changed. A model based approach would be easier to tune, since it would be able to provide new gains based on the new parameters.

8.2 Future work

As the system is still in the early stages, there is plenty of future work to be carried out on the project.

Mechanical work includes replacing the ball joint, as over time the connection from the ball to the car has become loose. The drive wheels are also misaligned, so either some kind of calibration can be done to balance them, or they can be replaced with thicker wheels. The material of the wheels contact area would also benefit from some form of rubber coating. A counterbalance placed on the back of the cart would stop the cart toppling forward when the motors are off. The reflective sensor plate and cone could be replaced with an alternative which allows the wires to pass through cleanly. The wires in general can be cut to more uniform lengths and groups stuck together with something more binding than cable ties.

The IR reflective distance sensors can be used once the wires have been cleaned up and the plate replaced. Filtering methods can then be implemented to include sensor readings with those by the DMP, the result would be very accurate readings. Redesign of the reflective plate and base cone should take into account the fact that black material does not reflect IR waves; a property which could be exploited.

There are settings to configure in the HC-05's AT command mode, these are worth exploring to improve performance. A new method of interfacing with Q-TIP could be worth researching. The app works for driving the car, but has its limitations. It has a low signalling rate for input which could be remedied with a new module, possibly communicating over wireless instead of Bluetooth. It also has a limited command range, specifically 10 digital commands; the app interface could be replaced with a joystick for an analog input range.

Q-TIP had trouble driving and stabilising the pendulum at the same time. There were two reasons for this. Reason one being the MPU6050 picked up readings from movement of the cart. An additional form of sensing on the cart would help isolate the cart movements readings from the pendulum movement readings. Reason two being the HC-05 had trouble receiving commands from the app during the I²C communication of the ESCs. A different Android device can be tested to see if it was device specific, or an app can be created using different software.

Parameter estimation is essential in the next steps, as the simulations can then be used to test controllers before being uploaded to Q-TIP. A model based approach for control design can then produce a range of more sophisticated controllers such as H_{\inf} which is robust to disturbances. Then it may also be viable to rotate the motor mounts see if yaw references can be tracked.

References

- [1] F. Grifo, “Design and construction of quadrotor top inverted pendulum (q-tip),” 2018.
- [2] C. Technologies, “Md10c enhanced 10 amp dc motor driver user’s manual rev.20,” 2013, [Accessed: Sep. 25, 2018]. [Online]. Available: <https://www.robotshop.com/media/files/PDF/user-manual-md10c-v2.pdf>
- [3] dfrobot, [Accessed: Sep. 25, 2018]. [Online]. Available: [https://www.dfrobot.com/wiki/index.php/12V_DC_Motor_251rpm_w/Encoder_\(SKU:_FIT0186\)](https://www.dfrobot.com/wiki/index.php/12V_DC_Motor_251rpm_w/Encoder_(SKU:_FIT0186))
- [4] Dejan, “Arduino and hc-05 bluetooth module tutorial,” 2016, [Accessed: Sep. 25, 2018]. [Online]. Available: <https://howtomechatronics.com/tutorials/arduino/arduino-and-hc-05-bluetooth-module-tutorial/>
- [5] Arduino, 2015, [Accessed: Sep. 25, 2018]. [Online]. Available: <https://www.arduino.cc/en/Tutorial/MultiSerialMega>
- [6] P. R. . Electronics, [Accessed: Sep. 25, 2018]. [Online]. Available: <https://www.pololu.com/product/2474>
- [7] A. Sanjeev, “How to interface arduino and the mpu 6050 sensor,” 2018, [Accessed: Sep. 25, 2018]. [Online]. Available: <https://maker.pro/arduino/tutorial/how-to-interface-arduino-and-the-mpu-6050-sensor>
- [8] L. Zanella, [Accessed: Sep. 25, 2018]. [Online]. Available: <http://lucaszanella.com/>
- [9] P. Uhing, “Design, modeling, and control of a two degree of freedom pendulum on an omnidirectional robot,” *Graduate Theses and Dissertations*, 2016.
- [10] J. Rowberg, [Accessed: Sep. 25, 2018]. [Online]. Available: <https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050>
- [11] Debra, “Mpu-6050 redux: Dmp data fusion vs. complementary filter,” [Accessed: Sep. 25, 2018]. [Online]. Available: <http://www.geekmomprojects.com/mpu-6050-redux-dmp-data-fusion-vs-complementary-filter/>
- [12] Aqibidris, “Sharp gp2y0a02 – calibration with arduino,” 2016, [Accessed: Sep. 25, 2018]. [Online]. Available: <http://www.diyspacepk.com/sharp-gp2y0a02-calibration-with-arduino/>
- [13] R. I. Content Development Team, [Accessed: Sep. 25, 2018]. [Online]. Available: <https://roboindia.com/tutorials/arduino-hc-05-at-mode>
- [14] B. Earl, “Calibrating sensors,” 2018, [Accessed: Sep. 25, 2018]. [Online]. Available: <https://cdn-learn.adafruit.com/downloads/pdf/calibrating-sensors.pdf>
- [15] DSSCircuits, [Accessed: Sep. 25, 2018]. [Online]. Available: <https://github.com/DSSCircuits/I2C-Master-Library>
- [16] A. J. Brizard, “An introduction to lagrangian mechanics,” 2014.