

SBCA

Self-Balancing Cube of Awesomeness



Presented by:
Griffin Trace

Prepared for:
Dr Francois Schonken
Dept. of Electrical and Electronics Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town
in partial fulfilment of the academic requirements for a Bachelor of Science degree in
Electrical and Computer Engineering.

October 27, 2025

Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced. Any section taken from an internet source has been referenced to that source.
3. This report is my own work, and is in my own words (except where I have attributed it to others).
4. I have not paid a third party to complete my work on my behalf. My use of artificial intelligence software has been limited to grammar checking.
5. I have not allowed and will not allow anyone to copy my work with the intention of passing it off as his or her own work.
6. I acknowledge that copying someone else's assignment or essay, or part of it, is wrong, and declare that this is my own work.

Signed: Griffin Trace 

Word Count: 14659

Acknowledgments

I'd like to thank my supervisor, Dr. Francois Schönken, for his guidance and patience throughout this project. His practical advice and clear direction helped keep the Self-Balancing Cube of Awesomeness from toppling over both literally and figuratively.

Thanks to my friends, the UCT Tour Guides, for their endless support, humour, and late-night working sessions.

A huge thanks to my family for always being in my corner. Special mention goes to my mom for her rigorous eye in proofreading this report... any remaining mistakes are entirely my own.

Lastly, thanks to the UCT Department of Electrical Engineering for providing the tools and space needed to bring this project to life.

Abstract

This project presents the design, modelling, simulation, and construction of the Self-Balancing Cube of Awesomeness (SBCA), a dynamically unstable robotic system capable of maintaining equilibrium on its edges or corners through the use of three internal reaction wheels. Inspired by ETH Zürich's Cubli, the SBCA aims to demonstrate advanced control theory concepts using low-cost and accessible components. The project encompasses analytical modelling of the cube's dynamics, hardware selection, and control system design, employing a linear quadratic regulator (LQR) for stabilisation. A despool-tilt controller was implemented to manage motor saturation and maintain stable wheel speeds, ensuring reliable performance using open-loop ESCs within strict budgetary constraints. MATLAB and Simulink were used extensively for simulation and validation of jump-up and balancing behaviours. The physical prototype was constructed using 3D-printed carbon-fibre-reinforced components and an ESP32 microcontroller for real-time control. Experimental results show successful balancing and confirm that the proposed design can achieve the required torque and speed characteristics for jump-up and corner balance. The SBCA demonstrates the feasibility of implementing complex nonlinear control on a constrained embedded system and provides a valuable educational and experimental platform for studying dynamic stabilisation through angular momentum exchange.

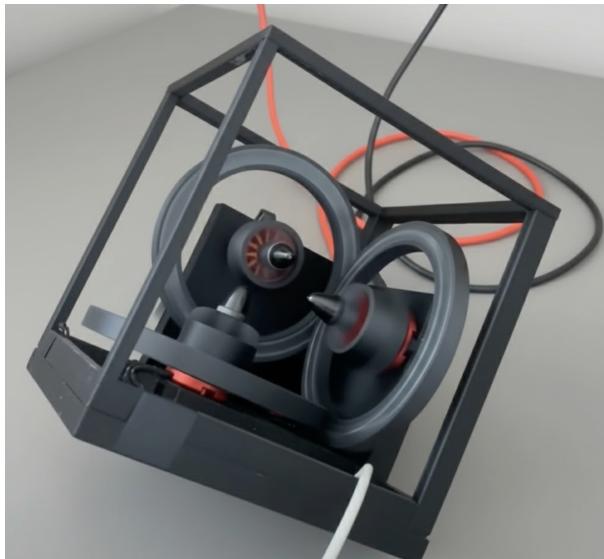


Figure 1: SBCA Balancing

Contents

1	Introduction	1
1.1	Background to the study	1
1.2	Objectives of this study	1
1.2.1	Problems to be investigated	1
1.2.2	Purpose of the study	2
1.3	Scope and Limitations	3
1.3.1	Scope	3
1.3.2	Limitations	3
1.3.3	GitHub Repository	4
1.4	Plan of development	4
2	Literature Review	5
2.0.1	Control System	5
2.0.2	Angle and Rotational Measurements	6
2.0.3	Jump Up	6
2.0.4	Physical Construction	7
2.0.5	Reaction Wheel Motors	8
2.0.6	Controller	8

3 Methodology	10
3.1 Overview	10
3.2 System Requirements	11
3.3 Specifications	12
3.4 Analytical Analysis	13
3.4.1 Assumptions and values	13
3.4.2 Energy Required to ‘jump up’	13
3.4.3 Torque Requirements	18
3.4.4 Motor Requirements Summary	19
3.4.5 Natural Frequency	19
3.5 Initial Simulation	20
3.5.1 Model Setup	21
3.5.2 Edge Jump	22
3.5.3 Edge Jump Simulation Results	22
3.5.4 Corner Jump	23
3.5.5 Corner Jump Simulation Results	25
3.5.6 Simulation Conclusion	26
3.6 Hardware Design	26
3.6.1 Motor	26
3.6.2 Motor Controller	29

3.6.3	Cube	34
3.6.4	Reaction Wheel	38
3.6.5	IMU	38
3.6.6	Microprocessor	39
3.6.7	Library Issues	40
3.6.8	Power Supply	41
3.6.9	Additional Features and Aesthetic Design	42
3.7	Bill of Materials (BOM)	42
3.8	LQR Controller Design	43
3.8.1	Rotational Dynamics of the Cube	43
3.8.2	Linear Quadratic Regulator (LQR)	44
3.8.3	Angle-Dependent Gain Blending	45
3.8.4	State Vector from Rotation Matrices	46
3.8.5	Torque Saturation and Smoothing	46
3.9	Despool Tilt Controller	47
3.10	Simulation	49
3.10.1	Model Setup	50
3.10.2	Controller Implementation	50
3.10.3	Motor and Actuator Modelling	50
3.10.4	Simulation Objectives	51

4 Results	52
4.1 Motor Results	52
4.2 Simulation Results	56
4.2.1 Motor Simulation	56
4.2.2 Edge Jump, and Balance Simulation	56
4.2.3 Corner Jump, and Balance Simulation	58
4.2.4 Simulation Results Conclusion	59
4.3 Balancing Results - Physical Testing	60
4.3.1 Edge Balancing	60
4.3.2 Corner Balancing	61
4.4 Results Conclusion	63
5 Discussion	65
6 Recommendations	67
7 Conclusions	69
A Addenda	73
A.1 Ethics Forms	75

Chapter 1

Introduction

1.1 Background to the study

Modern control systems increasingly deal with unstable and nonlinear dynamics, where balance and stability must be achieved through continuous feedback control. Self-balancing systems provide an accessible platform to study such principles, demonstrating how sensors, actuators, and controllers interact in real time.

Inspired by research such as the Cubli from ETH Zürich, this study explores how similar balancing behaviour can be achieved using low-cost components. The motivation is to make advanced control theory more practical and accessible by developing a compact, affordable system that demonstrates dynamic stabilisation through feedback control.

1.2 Objectives of this study

The primary goal of this study is to successfully model, simulate, design, construct, and test a robust feedback control system to allow a cube to self-balance on one of its corners.

1.2.1 Problems to be investigated

To achieve the primary goal, the following problems need to be solved

- **System Modelling:** Design an accurate and robust model of the system, including the cube's dynamics (such as inertia and gravity), control mechanisms (such as reaction wheels), as well as other dynamics(such as the non-linearity of the control motors used). This model will be used to simulate the self-balancing cube and to

test different control strategies.

- **Controller Design:** Design and tune an appropriate high-performance feedback control system to stabilise the self-balancing cube on its inherently unstable equilibrium point of its corner, or edge. This controller must ensure robust and stable control in real time, while rejecting small disturbances.
- **Initial State Management:** Develop a strategy to transition the cube from an initial stable state (lying flat on one of its faces) to its final desired state (such as on its corner or edge).
- **Physical Model:** Development of the physical prototype, including integration of the inertial measurement unit (IMU), motors, flywheels, power supply, and microcontroller.

1.2.2 Purpose of the study

The purpose of this study is as follows:

- **Proof of Concept:** The proof of concept will provide a practical demonstration of the abilities and dynamics of a stability system using angular momentum exchange. Although there may be limited direct practical uses of a self-balancing cube, the proof of concept will provide a valuable demonstration of concepts, such as reaction and control systems used in the aerospace industry.
- **Validation of Control Theory:** Provide a real-world, complex platform to validate high-complexity feedback control topics, reinforcing theoretical understanding, combined with practical implementation challenges.
- **Educational Demonstration:** The study contributes to a growing body of educational demonstrations that inspire curiosity and promote deeper engagement with modern control engineering.

1.3 Scope and Limitations

1.3.1 Scope

The scope of this study is limited to the design, modelling, and implementation of a Self-Balancing Cube. The project focuses on the following aspects:

- **Hardware and Construction:** Design and fabrication of the cube's structure, integrating all necessary components, including the microcontroller, IMU, motors, reaction wheels, and power supplies.
- **Modelling and Simulation:** Development of a comprehensive model of the system suitable for use in control simulation and design environments.
- **Control Implementation:** Design and implementation of feedback control algorithms, and their implementation on the chosen microcontroller platform.
- **Experimental Validation:** The validation of the control algorithms and hardware in the prototype system.

1.3.2 Limitations

The following factors may influence the depth and breadth of this study:

- **Time Constraints:** The project timeline is limited to a single academic semester, which restricts the time available for iterative design, comprehensive tuning, optimisation of hardware beyond proof-of-concept requirements, or exploration of advanced nonlinear control techniques.
- **Budget:** The study is limited to a budget of R2000. This limits the hardware to low-cost, readily available components, which will introduce limited functionality, resolution, noise and other design challenges.
- **Model Fidelity:** While an accurate model and simulation of the cube will be made, additional complexities such as friction, air resistance and other complexities will be excluded.

1.3.3 GitHub Repository

All related work, simulations, firmware and CAD models can be found in the following GitHub Repository: SBCA GitHub

1.4 Plan of development

This report is organised into several chapters, each addressing a key aspect of the design and development of the Self-Balancing Cube of Awesomeness (SBCA).

chapter 1 introduces the background to the study, outlining the motivation for the project and its relevance within the field of control systems engineering.

chapter 2 presents a review of the relevant literature, discussing previous work on self-balancing systems, control strategies, and related implementations.

chapter 3 details the methodology followed, including analytical modelling, hardware design, simulation, and testing procedures.

chapter 4 presents and analyses the results obtained from simulation and experimental testing.

chapter 5 provides a discussion of the results in relation to theoretical models and prior research.

chapter 6 outlines recommendations for future improvements and possible extensions of the system.

And finally chapter 7 concludes the report by summarising the main findings and confirming how the project objectives were achieved.

Chapter 2

Literature Review

A self-balancing cube is a dynamically unstable robotic system that maintains equilibrium, balanced on one of its corners or edges through the application of controlled reaction torques [1]. It represents a three-dimensional extension of the classic inverted pendulum problem, offering a rich platform for exploring nonlinear dynamics and control [1, 2]. A device such as this offers an ideal test bed for control systems research and teaching [3]. Several notable projects have previously tackled this challenge, most famously the Cubli developed at ETH Zürich in 2012 [4]. While these designs have gained significant attention, and ETH Zürich remains prominent in self-balancing cube research, a wide range of alternative methods, control strategies, and modelling approaches have been proposed. This review examines previous works on the topic to provide context and insight into techniques used to address the complexities of a self-balancing cube design.

2.0.1 Control System

Several different methods can be used to apply the forces and torques required to balance the cube. The most common approach is to use three reaction wheels, each applying a torque on one of the cube's three axes [1]. This is the method used in the design of Cubli [4], and is used across the board, from ReM-RC's 3D printed DIY Self-Balancing-Cube [5], to NIKOLA TOY's self-balancing cube [6], with several academic papers focusing on the control schemes and dynamics of this system [7, 1, 2].

In an alternative to the use of three reaction wheels, ETH Zürich previously developed a ‘Balancing Cube’ using six counterweights moved by a control system to maintain the cube’s centre of mass above the pivot point [8]. This ‘Balancing Cube’ is significantly larger and more complex than Cubli or other designs using reaction wheels [4].

Another approach is that of M. Hofer et al. also from ETH Zürich, being their ‘One-Wheel Cubli’ [9]. This device takes advantage of different rotational moments of inertia in the pitch and roll directions, allowing it to be controlled with just one reaction wheel

[9]. To obtain the differences in moments of inertia, this design uses a carbon fibre rod, with two counterweights extending out of the cube [9]. Although extremely novel, this method is challenging to fit into a conventional cube form factor.

2.0.2 Angle and Rotational Measurements

For the cube to balance on its corner or edge, the cube's controller must know its current position in order to calculate the actions it must take to move towards a balanced position.

The designs originating from ETH Zürich, the ‘Cubli’ [4], the ‘One-Wheel Cubli’ [9], and the ‘Balancing Cube’ [8] all make use of six IMU sensors. The data from the six IMU accelerometers is combined to accurately determine the orientation of the cube, using a method described by S. Trimpe [8]. The use of multiple accelerometers allows the direction of gravity to be determined, and thus the orientation, while ignoring accelerations detected by the sensors as a result of the cube’s rotation [8]. Although this can determine the cube’s orientation with high accuracy, the use of six IMUs adds complexity and cost, and they need to be very precisely positioned [8, 10]. F. Bobrow builds on the work of the Cubli, exploring the use of quaternions for control calculations, and only requiring one IMU [10]. The angular accelerations read from the IMU gyroscope are integrated to determine the orientation, although this is prone to drift [8]. So F. Bobrow determines the orientation using the IMU accelerometer as well, and although not accurate over the short term, it can be used to slowly correct the gyroscopes’ drift over the longer term [10]. This can be done using Mahony [11] or Madgwick [12] filters. These filters further combine the accelerometer and gyroscope data with magnetic sensor data to obtain more accurate results than can be obtained with only accelerometer and gyroscope data and a Kalman filter [11, 12]. The use of one IMU is seen in many of the self-balancing cubes originating outside of ETH Zürich [1, 2, 5, 13], and provides satisfactory data [10].

2.0.3 Jump Up

One of the goals for the self-balancing cube design is for the cube to be able to jump up or stand up onto an edge or corner on its own. The primary method used to achieve this is that laid out by the Cubli design team [3], entailing spinning up the reaction wheels to a desired speed and then braking them [3]. Through the conservation of angular momentum, the momentum stored in the reaction wheels is transferred to the cube [3]. Since the torque required for the braking is relatively high, much larger than most motors

can generate [13], a brake controlled by a servo is typically used [3]. The brake mechanism on the Cubli is a metal plate that is moved into the path of a bolt on the reaction wheel by a servo motor [3]. Alternatively, T. Liao et al. make use of a servo-controlled brake pad [1].

When the brake is applied, it is assumed to be an inelastic collision, so the conservation of angular momentum can be used to determine the cube's momentum after the braking [3]. Using the cube's momentum and the conservation of energy, and examining how much potential energy the cube will gain by standing up, it is possible to determine the reaction wheels' required momentum, and their required angular velocity [3, 1, 2].

2.0.4 Physical Construction

A variety of construction methods are used for the different cubes. From CNC cut carbon fibre [9], laser cut aluminium sheets connected with 3D printed brackets [10], fully CNCed aluminium frames[13], to a fully 3D printed cube [5]. One key requirement is to reduce the weight of the cube, in order to reduce the force required for the cube to jump up, and increase the angle and disturbances the cube can recover from [4]. However, the frame still needs to be strong enough to withstand the forces it experiences, especially during the jump up [4]. The original Cubli is a 15 cm box [4], which has become fairly standard for other self-balancing cube designs [5, 2, 13].

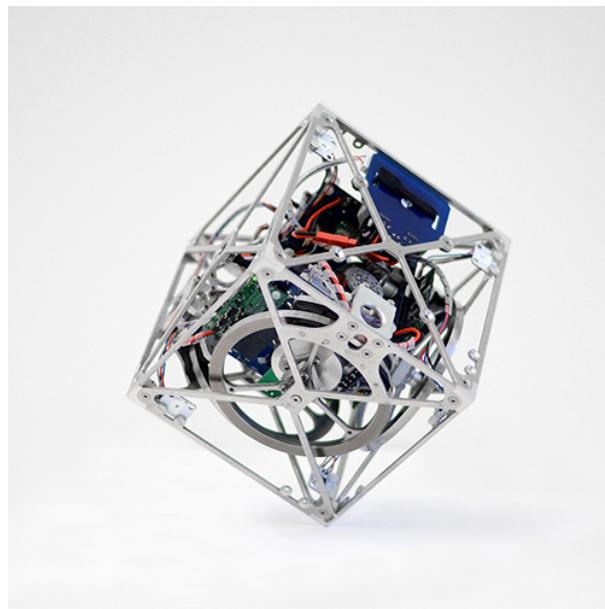


Figure 2.1: Picture of ETH Zürich's Cubli

Source: idsc.ethz.ch

For reaction wheels, most designs use laser-cut or CNC-cut aluminium [4, 2, 13]. One exception is the fully 3D-printed cube by Rem-RC, which makes use of 3D-printed reaction wheels, with nuts and bolts added for extra weight and inertia [5]. When using a braking system such as that used in Cubli, increasing the inertia of the wheels helps reduce the angular velocity required to perform the jump up; however, if taken to an extreme, this can also reduce the cube’s ability to recover from disturbances while balancing [3].

2.0.5 Reaction Wheel Motors

The Cubli uses a Maxon Motor AG 50 W EC-45 Flat to drive its reaction wheels [3, 4]. This brushless DC motor features a kV rating of 324 rpm/V, delivers a nominal torque of 0.112 Nm, a stall torque of 1.19 Nm, and operates at a nominal maximum speed of 4690 rpm [14]. Similar self-balancing cube implementations use comparable motor alternatives [13, 10]. For instance, the Rem-RC system utilises a Nidec CMC-24H motor [5], which is functionally similar but less powerful [15]. Likewise, T. L. Liao et al. employ a comparable but weaker motor in their design [1]. Overall, these systems commonly use 10–60 W brushless DC motors controlled via field-oriented control (FOC) techniques.

2.0.6 Controller

The control of a self-balancing cube is a challenging task due to its nonlinear and inherently unstable dynamics. As a result, a wide range of control strategies has been explored by various researchers. The original Cubli implementation employs a linear quadratic regulator (LQR), which provides a systematic framework for minimising a cost function that penalises deviations in angle and angular velocity [4, 3]. This approach makes use of a state space model of the cube, [2]. The use of LQR remains the most common approach across the literature, demonstrating strong performance in maintaining balance despite the assumption of local linearization around the upright equilibrium [16, 2]. Subsequent developments of the Cubli system investigated nonlinear control methods, notably a PD-like controller that incorporates wheel velocities and accelerations when computing control outputs [17]. This approach has been adopted by several other researchers, often decomposing the balancing problem into three independent axes, each governed by a dedicated PID controller [1, 13]. Beyond these conventional methods, alternative nonlinear strategies have also been explored. R. Singh et al. implemented model predictive control (MPC) to address system constraints and predict future states [18], while Z. Chen et al. investigated sliding mode control (SMC) to enhance robustness

against model uncertainties and external disturbances [7].

Conclusion

The literature reviewed demonstrates that self-balancing cubes are a rich platform for exploring nonlinear dynamics, control strategies, and mechanical design challenges. Reaction wheels remain the most widely used method for applying balancing torques, while alternatives such as counterweights or single-wheel designs highlight innovative approaches to reducing system complexity. Accurate angle and rotational measurements are critical, with multiple IMUs or single IMU fusion techniques providing reliable orientation data. Jump-up capabilities require careful consideration of reaction wheel inertia and braking mechanisms, while the choice of motors and materials significantly impacts performance and robustness. Control strategies range from linear methods such as LQR to nonlinear and predictive approaches, with each offering trade-offs between simplicity, accuracy, and robustness. Overall, previous research provides a strong foundation for designing a functional and responsive self-balancing cube, guiding both mechanical and control system development.

Chapter 3

Methodology

This chapter outlines the methodology employed to design, simulate, construct, and test the Self-Balancing Cube of Awesomeness, (from here referred to as SBCA). The design process followed an iterative and analytical approach. Initially, the system was modelled and analysed through both theoretical derivation and simulation, using reasonable assumptions to establish the hardware specifications. Based on these results, the hardware and electrical subsystems were designed to satisfy the identified specifications. The selected components were then modelled and re-simulated to verify their suitability for the intended application. Following this, the control system was designed and simulated to ensure that all performance requirements and design objectives were met. Finally, the physical cube was assembled, programmed, and experimentally tested, with results and observations recorded for validation and analysis.

3.1 Overview

The SBCA project aims to achieve stable attitude control using three internal reaction wheels, a design choice made after an exploration of relevant literature and inspired by successful dynamic balancing systems, such as the Cubli [4]. The development process was broken down into four main stages:

1. Preliminary simulation and modelling to define specifications.
2. Hardware design and component selection.
3. Control system design and simulation.
4. Embedded software implementation, calibration, and testing.

3.2 System Requirements

Table 3.1: User, functional, and design requirements of the Self-Balancing Cube of Awesomeness (SBCA).

Requirement ID	Description
Functional Requirements	
FR1	The SBCA must be capable of maintaining balance on its edges without external support.
FR2	The SBCA must be capable of maintaining balance on a corner using coordinated control of its reaction wheels.
FR3	The SBCA must recover from small external disturbances and return to an upright position.
FR4	The SBCA must include a mechanism to reduce accumulated wheel speed during prolonged operation (despooling).
FR5	The SBCA must operate as a fully self-contained unit, with all electronic, mechanical, and power components enclosed within the cube.
FR6	The SBCA must include at least one additional feature that demonstrates its “Awesome” characteristic (e.g., lighting or motion effect).
Software and Control Requirements	
SR1	The firmware must implement closed-loop attitude control using feedback from the IMU.
SR2	The control system must operate continuously and in real time to maintain stability.
SR3	The firmware must include safety limits to prevent reaction wheel saturation or overcurrent conditions.
Safety and Reliability Requirements	
SAF1	The SBCA must perform a safe shutdown in the event of a critical error or low battery voltage.
SAF2	The system must prevent overheating and protect components from excessive current draw.
Design and Project Requirements	
DR1	The total cost of the project must remain within the allocated budget.
DR2	The complete design, construction, and testing process must be completed within one academic semester.
DR3	The cube’s physical dimensions must remain compact and symmetrical to maintain aesthetic and functional balance.
DR4	All designs, simulations, and firmware must be documented and stored in a version-controlled repository.

The requirements listed above define the core behavioural, functional, and project expectations for the SBCA.

3.3 Specifications

The requirements form the foundation from which the specifications were derived. Based on the requirements, preliminary analytical analysis and simulations were conducted to determine the design parameters and performance specifications of the SBCA. The details of this process are given in section 3.4 and section 3.5 of this report.

Table 3.2: Functional, control, safety, and design specifications of the Self-Balancing Cube of Awesomeness (SBCA).

Specification ID	Description
Functional Specifications	
FS1	Capable of balancing on a single edge for at least 30 s without external intervention.
FS2	Capable of balancing on a single corner for at least 30s.
FS3	Performs a jump-up manoeuvre from a resting face to an edge or corner.
FS4	Can reject external disturbances, gentle taps, without toppling.
Control and Software Specifications	
SS1	Controller sampling frequency: $> 500\text{Hz}$
SS2	Control latency (IMU to actuator output): $< 5\text{ms}$
SS3	Gyroscope full-scale range: $\pm 2000^\circ/\text{s}$
SS4	Maximum wheel angular speed: 10000rpm
SS5	Reaction wheel torque capability: $\approx 0.3\text{Nm}$
Safety and Reliability Specifications	
SAFS1	Motor current limit: 30A
SAFS2	Power source: 3S LiPo battery (12.6V)
SAFS3	Maximum cube temperature during operation: $< 50^\circ\text{C}$.
SAFS4	System includes a software fail-safe to cut motor power if motor speed is over 10000 rpm.
SAFS5	All rotating components enclosed to ensure operator safety.
Design and Hardware Specifications	
DS1	Cube side length: $\approx 100\text{mm}$
DS2	Total mass: $\approx 0.5\text{kg}$ (including electronics and reaction wheels).
DS3	Reaction wheel mass: $\approx 50\text{g}$
DS4	Total estimated cost: $< \text{R}2000$

Several of these specifications are soft specifications, acting as a guideline during the design process.

3.4 Analytical Analysis

In order to help determine the specifications for the hardware of the SBCA, an analytical analysis of the system was performed based on a number of simplifying assumptions. These calculations provided estimates for inertia, required torques, and the dynamics of the cube. The results were used to set preliminary targets for specifications such as motor torque, wheel inertia, and control loop bandwidth.

3.4.1 Assumptions and values

To begin a number of assumptions and reasonable estimates were made. These values will be confirmed and updated once the final design of the cube has been completed, as detailed in section 3.6. The initial assumptions were that the cube has an edge length of 10 cm, and the total mass is 500 g. For simplicity, the mass is assumed to be uniformly distributed. The reaction wheels are assumed to have an outer diameter of 10 cm, and an inner diameter of 8 cm, with a weight of 50 g, resulting in a moment of inertia of $1.02510^{-4} \text{ kgm}^2$.

Table 3.3: Analytical assumptions and key parameters

Parameter or assumption	Value / note
Cube edge length, a	0.10 m
Total mass, m	0.50 kg
Mass distribution	Uniform
Gravitational acceleration, g	9.81 ms^{-2}
Cube moment of inertia about centre, I_{cm}	$I_{cm} = \frac{1}{6}ma^2$
Cube moment of inertia about edge pivot, I_{edge}	$I_{edge} = I_{cm} + md^2$
Reaction wheel moment of inertia, I_{wheel}	$1.02510^{-4} \text{ kgm}^2$
Small-angle linearisation	$\sin \theta \approx \theta$ for $ \theta \ll 1$ (rad)

3.4.2 Energy Required to ‘jump up’

To jump the cube up onto an edge

Using the assumptions in Table 3.3, the SBCA will have a centre of mass in the centre of the cube, or 5 cm above the ground when lying on a face. To determine the energy needed to move the cube onto its edge, we analyse how much potential energy must be added to the system. First, we need to determine how much the centre of mass is raised.

3.4. ANALYTICAL ANALYSIS

When the cube is on its edge, the centre of mass will be between its top edge and its bottom edge. The distance between these two edges can be found using:

$$d = \sqrt{a^2 + a^2} = \sqrt{2}a$$

We know the centre of mass is midway between these, and so the centre of mass is at a height of:

$$h = \frac{\sqrt{2}}{2}a$$

For a cube with edge length $a = 0.1$ m:

$$h = \frac{\sqrt{2}}{2} \cdot 0.1 \approx 0.0707 \text{ m}$$

Since the starting position (cube lying on a face) is at $h_{initial} = 5 \text{ cm} = 0.05 \text{ m}$, it has gained:

$$\Delta h = h - h_{initial} = 0.0707 - 0.05 = 0.0207 \text{ m}$$

The energy gained by raising the centre of mass can be calculated using the gravitational potential energy formula:

$$E_{gain} = mg\Delta h$$

We have assumed the mass of the cube is $m = 0.5 \text{ kg}$ and $g = 9.81 \text{ ms}^{-2}$:

$$E_{gain} = 0.5 \cdot 9.81 \cdot 0.0207 \approx 0.101 \text{ J}$$

$E_{gain} \approx 0.10 \text{ J}$

3.4. ANALYTICAL ANALYSIS

The energy required to lift the cube's centre of mass (stand the cube up on its edge) will be delivered from the reaction wheels' rotational kinetic energy. We can calculate the required angular momentum that needs to be transferred to the cube to lift it with:

$$E_{gain} = \frac{1}{2} I_{edge} \omega^2$$

Solving for the angular velocity ω :

$$\omega = \sqrt{\frac{2E_{gain}}{I_{edge}}}$$

Substituting $E_{gain} = mg\Delta h$ and $I_{edge} = \frac{2}{3} ma^2$:

$$\omega = \sqrt{\frac{2(mg\Delta h)}{\frac{2}{3}ma^2}} = \sqrt{\frac{3g\Delta h}{a^2}}$$

Finally, the inertial (angular) momentum needed is:

$$L = I_{edge} \omega = \frac{2}{3} ma^2 \sqrt{\frac{3g\Delta h}{a^2}} = \frac{2}{3} ma^2 \cdot \sqrt{\frac{3g\Delta h}{a^2}} = \frac{2}{3} ma \sqrt{3g\Delta h}$$

Substituting in the assumed values: $m = 0.5 \text{ kg}$, $a = 0.1 \text{ m}$, $\Delta h = 0.0207 \text{ m}$, $g = 9.81 \text{ ms}^{-2}$:

$$L = \frac{2}{3}(0.5)(0.1)\sqrt{3 \cdot 9.81 \cdot 0.0207} \approx 0.032 \text{ kg m}^2 \text{s}^{-1}$$

$L \approx 0.032 \text{ kgm}^2 \text{s}^{-1}$

The required angular momentum to lift the cube will be imparted from the angular momentum of a reaction wheel. If we treat the transfer of momentum between the cube and the reaction wheels as an inelastic collision, the principle of the conservation of momentum can be applied [3]. From this, we can determine the angular velocity the reaction wheel needs to have:

$$L_{wheel} = I_{wheel} \omega_{wheel}$$

Solving for ω_{wheel} , the required angular velocity of the wheel is:

$$\omega_{wheel} = \frac{L_{wheel}}{I_{wheel}} = \frac{0.032}{1.02510^{-4} kgm^2} \approx 312 \text{ rad/s}$$

Converting to revolutions per minute (rpm), the required wheel speed is:

$$rpm = \omega_{wheel} \cdot \frac{60}{2\pi} \approx 312 \cdot \frac{60}{2\pi} \approx 2979 \text{ rpm}$$

$\omega_{wheel} \approx 312 \text{ rad/s} \approx 2979 \text{ rpm}$

To jump up onto a Corner

We follow a similar process for the corner balancing scenario. When the cube is balanced on a corner, the space diagonal of the cube is vertical, and the centre of mass lies halfway along this diagonal. The height of the centre of mass above the ground is therefore:

$$h = \frac{\sqrt{3}}{2}a$$

For a cube of edge length $a = 0.1 \text{ m}$:

$$h = 0.05\sqrt{3} \approx 0.0866 \text{ m}$$

The cube initially rests with its centre of mass at $h_{initial} = 0.05 \text{ m}$. Thus, the vertical displacement of the centre of mass is:

$$\Delta h = h - h_{initial} = 0.0866 - 0.05 = 0.0366 \text{ m}$$

The corresponding increase in gravitational potential energy is:

3.4. ANALYTICAL ANALYSIS

$$E_{gain} = mg\Delta h = 0.5 \times 9.81 \times 0.0366 \approx 0.179 \text{ J}$$

Using the parallel axis theorem, the cube's moment of inertia about the corner pivot is:

$$I_{corner} = I_{cm} + md^2 = \frac{1}{6}ma^2 + m \left(\frac{\sqrt{3}a}{2} \right)^2 = 4.58 \times 10^{-3} \text{ kgm}^2$$

The angular velocity required for this energy is:

$$\omega = \sqrt{\frac{2E_{gain}}{I_{corner}}} = \sqrt{\frac{2(0.179)}{4.58 \times 10^{-3}}} \approx 8.85 \text{ rad/s}$$

The angular momentum needed is:

$$L = I_{corner}\omega = 4.58 \times 10^{-3} \times 8.85 \approx 0.0406 \text{ kgm}^2\text{s}^{-1}$$

If each reaction wheel has angular momentum magnitude L_w about its own axis, the component along the desired corner rotation axis is $L_w \cos 45^\circ$. Two wheels (mounted on adjacent faces at $\pm 45^\circ$) give a combined projection

$$L_{proj} = 2L_w \cos 45^\circ = 2L_w \frac{1}{\sqrt{2}} = \sqrt{2} L_w$$

Set this equal to the required total angular momentum L :

$$\sqrt{2} L_w = L \implies L_w = \frac{L}{\sqrt{2}}$$

Using $L = 4.06 \times 10^{-2} \text{ kg m}^2\text{s}^{-1}$ and $I_{wheel} = 1.025 \times 10^{-4} \text{ kg m}^2$:

$$L_w = \frac{4.06 \times 10^{-2}}{\sqrt{2}} \approx 2.8709 \times 10^{-2} \text{ kg m}^2\text{s}^{-1}$$

$$\omega_{wheel} = \frac{L_w}{I_{wheel}} \approx \frac{2.8709 \times 10^{-2}}{1.025 \times 10^{-4}} \approx 280.1 \text{ rad/s}$$

$$\text{rpm} = \omega_{wheel} \frac{60}{2\pi} \approx 2675 \text{ rpm (each)}$$

$$\omega_{\text{wheel}} \approx 280 \text{ rad/s} \approx 2675 \text{ rpm (each)}$$

Motor Speed Summary

The calculations above show that to jump the cube onto an edge, the reaction wheel will need a speed of 2979 rpm, and to jump onto a corner, two reaction wheels will need a speed of 2675 rpm. This is, of course, a severe simplification of the scenario and assumes the reaction wheels can impart all of their angular momentum instantaneously into the cube's body. The true required rpm will likely be significantly higher; however, this calculation provides a ballpark figure and order of magnitude of the rpm requirements for the motors.

3.4.3 Torque Requirements

Another aspect to consider is the minimum torque required to jump the cube up. Projects such as Cubli make use of a braking mechanism on the reaction wheels and so can generate extremely high torques. However, it may also be possible to generate the required torques directly from the motors.

By examining the torque the cube will experience due to gravity at the instant it begins to jump up, the minimum required torque for the jump can be determined: The torque from gravity is:

$$\tau_{g\text{Edge}} = mg \frac{a}{2}, \tau_{g\text{Corner}} = mg \frac{\sqrt{2}a}{2}$$

Using $m = 0.5 \text{ kg}$, $a = 0.10 \text{ m}$, $g = 9.81 \text{ ms}^{-2}$:

$$\tau_{g\text{Edge}} = 0.5 \cdot 9.81 \cdot 0.05 \approx 0.245 \text{ Nm}$$

and

$$\tau_{g\text{Corner}} = \sqrt{2} \cdot \tau_{g\text{Edge}} \approx 0.346 \text{ Nm}$$

$\tau_{g\text{Corner}}$ will be applied from two reaction wheels, each at $\pm 45^\circ$ from the direction of rotation, and so similarly to section 3.4.2

$$\tau_{g\text{CornerWheel}} = \tau_{g\text{Corner}} / \sqrt{2} \approx 0.245 \text{ Nm.}$$

3.4.4 Motor Requirements Summary

From the analytical analysis above, the following key findings were made:

- The minimum required motor speed is 2979 rpm, brequired to jump the cube up to its edge
- The minimum required motor torque is 0.245 Nm

Important to note is that both of these values assume the other is effectively infinite. For instance, a torque of just above 0.245 Nm would cause the cube to rise very slowly, meaning the torque must be applied over an extended period, causing a much higher wheel speed. Similarly, the angular velocity of 2979 rpm would require the entire angular momentum to be transferred instantaneously, demanding infinite torque. Hence, the actual values must be significantly greater in practice. While an analytical approach could estimate these interactions, the use of a simulation-based analysis was deemed more practical. A description of this is given below in section 3.5.

It is also worthwhile noting the relationship between the cube's parameters and these values.

- The rpm requirement is directly proportional to:
 - The mass of the cube
 - The length of the cube to the power of $3/2(a\sqrt{a})$
 - Inversely to the inertia of the reaction wheel.
- The torque requirement is directly proportional to:
 - The mass of the cube
 - The length of the cube

3.4.5 Natural Frequency

By considering the edge balancing scenario, we can determine a reasonable estimate of the cube's natural frequency, which is imperative in the controller design.

Considering small rotations θ about the edge pivot, the linearised rotational dynamics (including applied control torque τ_{rw} from the reaction wheels) are:

$$I_{edge} \ddot{\theta} = -mgd\theta + \tau_{rw},$$

where $-mgd\theta$ is the linearised gravitational restoring torque (for small θ).

Applying control law to the equation above, we obtain the natural frequency:

$$\omega_n = \sqrt{\frac{mgd}{I_{edge}}}.$$

Using the numerical values above:

$$mgd \approx 0.5 \times 9.81 \times 0.07071 \approx 0.3468 \text{ Nm},$$

$$\omega_n \approx \sqrt{\frac{0.3468}{0.003333}} \approx 10.2 \text{ rad/s},$$

which corresponds to a natural frequency $f_n = \omega_n/(2\pi) \approx 1.62 \text{ Hz}$ and a period $T \approx 0.62 \text{ s}$. The computed natural frequency ($\omega_n \approx 10.2 \text{ rad/s}$) suggests that a control-loop bandwidth significantly higher than the natural frequency is required for robust rejection of disturbances. These results motivated selecting a control update rate in the several hundreds of Hz for the SBCA controller.

3.5 Initial Simulation

Once the analysis was completed, a simulation was deemed the optimal approach to both verify the initial findings and gain a deeper understanding of the problem and its governing parameters. The simulation was conducted using the MATLAB Simulink Multibody toolkit. This environment is ideal as it allows for the complete modelling and simulation of the cube, including the forces acting on it and its complex interactions with itself and the ground, without the need for explicit derivation of the underlying mathematical equations.

3.5.1 Model Setup

The cube was modelled using the Multibody toolkit as a ‘Brick Solid’, with the dimensions and mass detailed in the analysis parameters (see Table 3.3). The internal reaction wheels were represented as disks with the specified inertia. These wheels are mounted to the main cube body using ‘Revolute Joints’, positioned perpendicular to each other to provide three-axis control. A second, stationary cube was introduced to function as the ‘ground’ on which the primary cube rests. Their physical interaction is simulated using a ‘Contact Force’ block, accurately modelling the forces between the cube and the surface.

A basic proportional-integral-derivative (PID) controller was implemented to minimise the error between the current and desired orientations. The controller’s parameters were determined via manual tuning and were recognised as non-optimal. While a more in-depth and thorough controller design was planned for a later stage (section 3.8), this simple PID loop was sufficient for the initial simulation and investigative phase of the cube’s dynamics. This controller outputs a torque, which is applied between the corresponding reaction wheel and the cube.

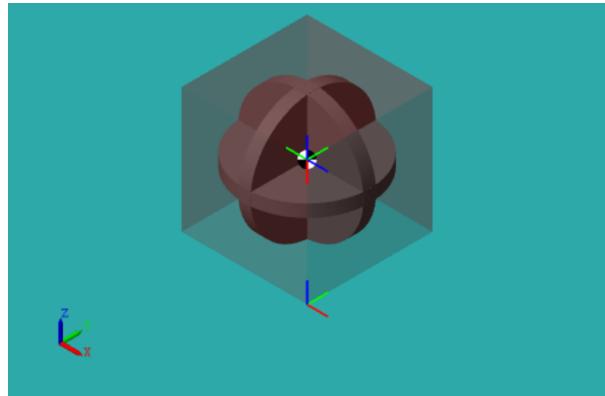


Figure 3.1: Graphical representation of the initial simulation, showing the cube and the three reaction wheels.

3.5.2 Edge Jump

Edge Orientation

At time equals 0.5 seconds, the desired set point was set to rotate the cube 45 degrees, corresponding to a rotational matrix of:

$$R_{\text{des}} = \begin{bmatrix} \cos(45^\circ) & 0 & \sin(45^\circ) \\ 0 & 1 & 0 \\ -\sin(45^\circ) & 0 & \cos(45^\circ) \end{bmatrix} = \begin{bmatrix} 0.707 & 0 & 0.707 \\ 0 & 1 & 0 \\ -0.707 & 0 & 0.707 \end{bmatrix}$$

3.5.3 Edge Jump Simulation Results

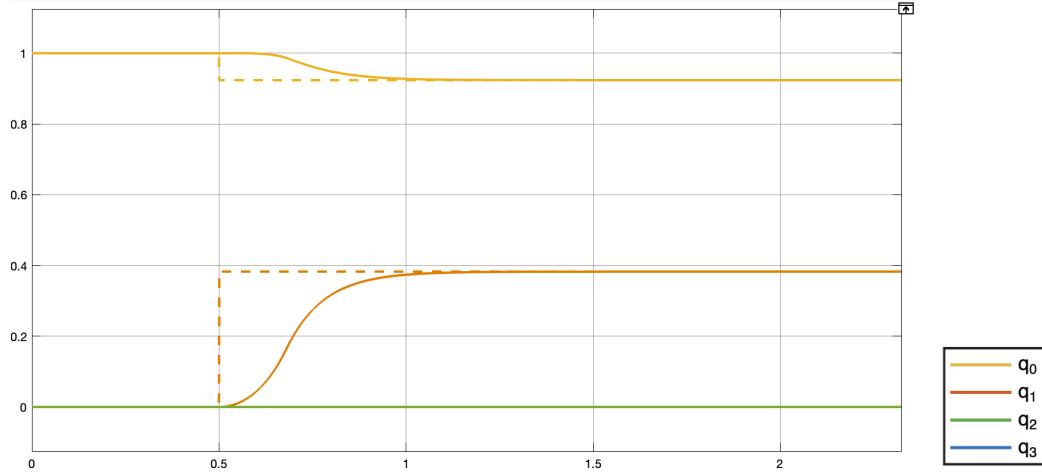


Figure 3.2: Graph showing cubes set point and position over time (s) (in quaternion form)

The controller simulation successfully applied torques to the reaction wheels, driving the cube into the desired orientation. Different parameters were experimented with, specifically the maximum torque the controller could generate, through which a better understanding of the dynamics was achieved.

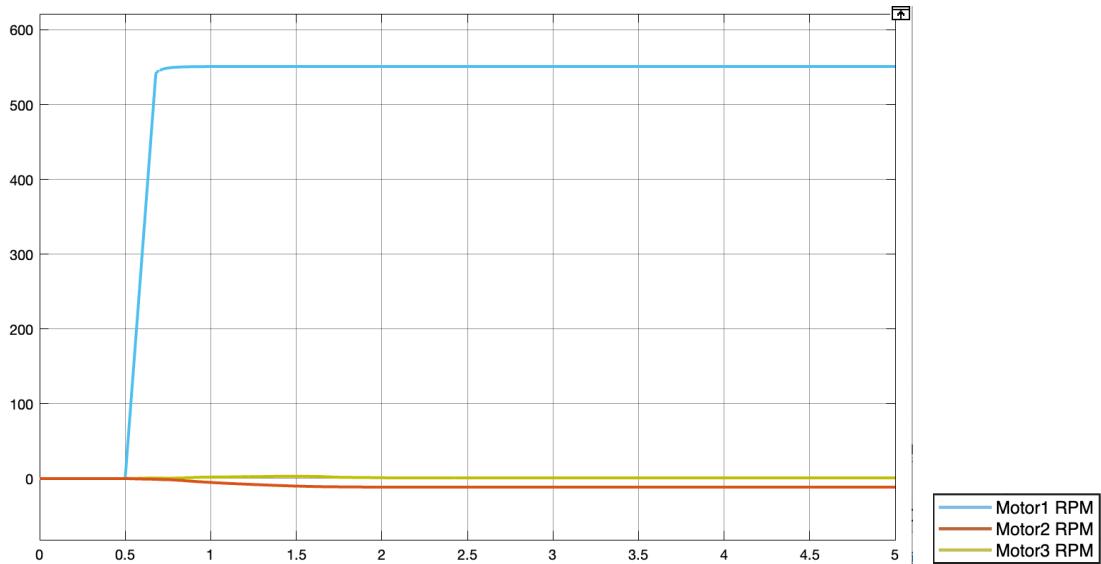


Figure 3.3: Graph showing reaction wheel speeds (rad/s) vs time (s) with a torque limit of 0.3 Nm

Figure 3.3 shows the reaction wheel speeds with a torque limit of 0.3 Nm. The simulation was repeated for several different torque limits to observe their effect on the cube's ability to rise to the edge position. The findings are summarised in Table 3.4.

Table 3.4: Edge jump performance for varying torque limits

Torque Limit [Nm]	Rise Time 10% to 90% [ms]	Peak Wheel Speed [rad/s]
0.20	—	Failed to 'jump up'
0.245	—	Failed to 'jump up'
0.246	330	1520
0.25	316	1080
0.30	254	551
0.35	237	456
0.40	229	410
1.00	216	380

3.5.4 Corner Jump

From section 3.4 it is expected that the corner jump will require the same minimum torque, but less angular velocity. In a similar way to the test with the cube jumping to the edge, the simulation experiment was repeated for the cube jumping up to its corner.

Corner Orientation

When the cube rests on a corner, its body diagonal $[1, 1, 1]^T$ is aligned with the global vertical axis. The goal is to construct the corresponding body-to-world rotation matrix, R_{corner} .

First, a unit vector is formed along the body diagonal:

$$\hat{z} = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

A helper vector $\mathbf{h} = [1, 0, 0]^T$ is chosen to define an orthogonal basis. The remaining unit vectors are then obtained from the cross products:

$$\hat{y} = \frac{\hat{z} \times \mathbf{h}}{\|\hat{z} \times \mathbf{h}\|}, \quad \hat{x} = \hat{y} \times \hat{z}.$$

Assembling these orthonormal vectors yields the rotation matrix:

$$R_{\text{corner}} = [\hat{x} \ \hat{y} \ \hat{z}] = \begin{bmatrix} 0.707 & -0.408 & 0.577 \\ 0 & 0.816 & 0.577 \\ -0.707 & -0.408 & 0.577 \end{bmatrix}.$$

This matrix represents the cube oriented such that one of its corners is in contact with the ground and the body diagonal is vertical. This is set as the desired orientation at 0.5 seconds into the simulation.

3.5.5 Corner Jump Simulation Results

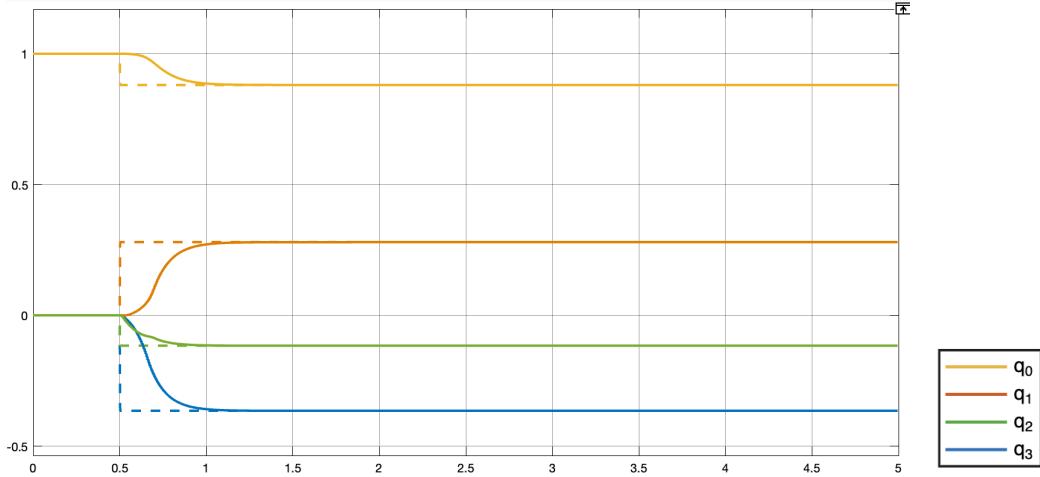


Figure 3.4: Graph showing cubes set point and position over time (s) (in quaternion form)

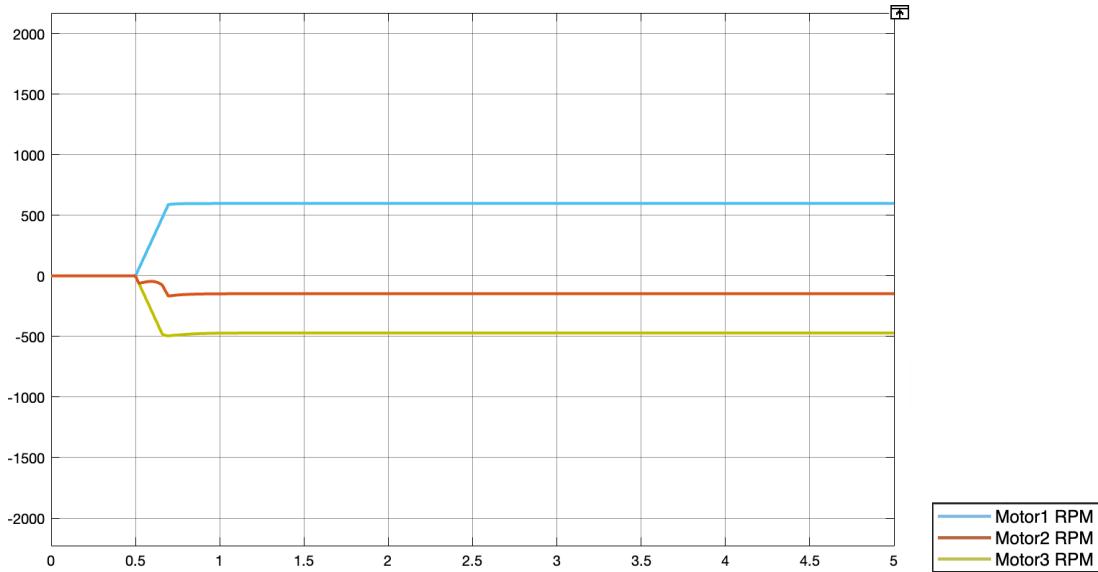


Figure 3.5: Graph showing reaction wheel speeds (rad/s) vs time (s) with a torque limit of 0.3 Nm

From Figure 3.4 it is clear that the cube again successfully moves to the desired set point, and so the same simulation experiment was completed, recording the peak wheel speeds needed at different torque levels.

Table 3.5: Corner jump performance for varying torque limits

Torque Limit [Nm]	Rise Time 10% to 90% [ms]	Peak Wheel Speed [rad/s]
0.20	—	Failed to 'jump up'
0.245	—	Failed to 'jump up'
0.246	295	928
0.25	285	900
0.30	251	597
0.35	238	506
0.40	230	461
1.00	215	360

3.5.6 Simulation Conclusion

From Table 3.4 and Table 3.5, it can be seen that our estimated minimum torque value in subsection 3.4.3 (0.245 Nm) appears to be correct. Additionally, we can see that as our torque value increases, our peak wheel speed approaches the value calculated in subsection 3.4.2(2979 rpm), and confirms the inverse relationship between the two. Finally, if we select a value of around 0.3 Nm of torque, the cube will require 551-597 rad/s, which is equivalent to 5261-5700 rpm. These are likely viable and achievable goals for our design, and these values were used as the motor specifications in section 3.3. This will be explored further in subsection 3.6.1.

3.6 Hardware Design

This section details the hardware design process of the SBCA.

3.6.1 Motor

Following the literature review and the initial investigation, it was decided that the best approach would be to select a motor for the reaction wheels and design the SBCA around it. From the analytical analysis and the initial simulation, we were looking for a motor that can provide roughly 0.3 Nm of torque and 5700 rpm. These were not hard specifications, as the exact requirements would depend on the final design. Variables like

the size and weight of the reaction wheels and the final size and weight of the cube's body have a large effect on the performance requirement. Therefore, the motor's performance must be verified to be adequate for the final design through simulation before proceeding with the construction of the SBAC.

A number of different motor options were explored, as detailed below. Key considerations are maximum torque and rpm, size, weight, and (it turned out), most importantly, price.

Summary of motor options

Option 1: Maxon Motor AG 50 W EC-45 Flat: The Cubli made use of this motor to drive its reaction wheels [3, 4]. This motor is a brushless DC motor with a built-in Hall effect sensor. In the ??, it was found that either this motor or similar models were used extensively across the board in previous projects. This motor provides high torque and adequate speeds, but is expensive and heavy. Due to its built-in Hall effect sensors, it can also easily be used with a field-oriented control (FOC) driver, providing excellent low-speed control.

Option 2: Nidec CMC-24H: This is a brushless DC motor with a built-in controller. It was used by Rem-RC in their design of the self-balancing cube [5]. This motor has a built-in controller with good low-speed performance and adequate top speeds. However, it provides low torque, and is heavy, and somewhat expensive.

Option 3: DC Brush Motor 6V: This motor is a simple brushed DC motor. It is readily available in South Africa, from stores such as Micro Robotics. It is extremely cheap and can be paired with cheap gearboxes to balance torque and speed. However, due to its low power rating, it simply will not be able to provide adequate torque and rpm.

Option 4: Nema 14 Stepper Motor: Commonly used and widely available, this motor will be able to provide good low-speed performance. However, it provides low torque, very low rpm, and is fairly heavy. Other similar stepper motors were also considered, but these disadvantages are present across the board.

Option 5: Mr. Steele V4 SILK 2306 (2345KV) 4S FPV freestyle motor: This drone motor can provide possibly adequate torque and exceptional speed. It is available locally and has a low mass. Its price is acceptable. However, being a drone motor, it will suffer from bad low-speed performance.

Option 6: DRN-2212 920KV Brushless Motor: Similar to option 5, but at a lower Kv rating, this drone motor provides adequate torque, speed and low mass at an excellent price. However, it will suffer from bad low-speed performance, and its available performance documentation is very inadequate.

Full Stats

The full stats of these motor options are given below:

Option	Max Torque(Nm)	Speed(rpm)	Weight(g)	Price(R)	Other
1	1.19	5740	116	≈ 3000	Native FOC support
2	0.07	5800	110	≈ 340	Built in controller
3	0.004	11500	18	≈ 8	Can be paired with a gearbox
4	0.09	≈ 1000	140	≈ 188	
5	≈ 0.20	39000	32	≈ 475	Bad low speed performance
6	≈ 0.31	10200	53	≈ 220	Bad low speed performance

Table 3.6: Motor stats

Decision

From Table 3.6 and the discussion above, it is clear that Option 6, the DRN-2212 920KV Brushless Motor, is likely the only viable option. Option 1, the Maxon Motor AG 50 W EC-45 Flat is probably a much better choice, providing much better torque, as well as, most importantly, native FOC support allowing excellent control at low speeds, but due to its extremely high price, it is not a viable option within the R2000 overall budget. However, the DRN-2212 920KV Brushless Motor meets the project's specifications and requirements, and its downfalls can likely be mitigated through clever design, and so it was chosen.

3.6.2 Motor Controller

With the selection of the DRN-2212 920KV Brushless Motor, an appropriate motor controller needed to be identified. The motor controller plays a critical role in the overall performance of the SBCA, as it directly governs the torque response, control bandwidth, and stability of the reaction wheels. The controller must ideally be capable of handling both high-speed operation and smooth low-speed torque generation.

From the analytical and simulation results, each motor must be capable of producing up to approximately 0.3 Nm peak torque, with steady-state wheel speeds up to roughly 6000 rpm. When paired with the DRN-2212, this means the controller must handle at least 30 A of current, at 6.5 V, based on the motor's Kv rating. However, it is intended that the SBCA will make use of a 3S LiPo battery, and so it must operate at a voltage range up to 12.6 V. In addition, the motor controller must interface with the microcontroller through a pulse width modulation (PWM) or serial control interface.

Options Considered

A number of electronic speed controllers (ESCs) and driver modules were considered for the DRN-2212 motor, focusing on local availability, control precision, and cost:

Option 1: BLHeli-S 30A Drone ESC: This low-cost 30A drone ESC is readily available in South Africa and compatible with the DRN-2212 motor. It provides excellent high-speed operation but lacks precise low-speed torque control, which is critical for balancing. This controller can be flashed with Bluejay firmware, which, through the use of Bi-Directional DShot (serial communication standard), allows the ESC to provide rpm telemetry back to the microcontroller. These ESCs can be found priced as low as R105, can handle the required current, and are lightweight.

Option 2: BLHeli 32 ESC: This ESC firmware supports advanced control features such as advanced open-loop speed regulation, telemetry, and bidirectional communication. While still designed primarily for drones, its 32-bit microcontroller and adjustable parameters allow better low-speed performance and control response. However, it does not provide true FOC, making it difficult to achieve smooth torque near zero speed. Generally, this is a better option than the BLHeli-S ESC. However, due to their use in global conflicts,

they have become nearly impossible to source, unfortunately ruling them out for use in this project.

Option 3: Makerbase SimpleFOC BLDC Controller: The SimpleFOC board allows full field-oriented control when paired with an encoder or Hall sensors. This approach provides direct control over the motor currents, yielding excellent low-speed torque characteristics. It also allows integration with custom firmware, enabling closed-loop torque commands directly from the balancing controller. Although this requires additional setup and sensor wiring, it would offer the best overall performance and tunability for the SBCA. Unfortunately, this module was ruled out due to its high cost of R435 per motor, as well as its peak current being limited to 5 A. Other SimpleFOC-based boards do exist, but none were found that would meet the current and price specifications.

Option 4: Makerbase ODrive BLDC Motor Controller: The ODrive is an industrial-grade brushless controller that provides advanced current, position, and torque control. Similar to the SimpleFOC board, this controller makes use of an encoder on the motor to allow precise closed-loop control. This board can provide 60 A of current at the required voltages. Unfortunately, it is fairly large and heavy, and very expensive, costing R1300, making it unviable for this project.

The problem

The selection of a motor controller became one of the biggest complexities of this project. The ODrive or SimpleFOC board are the only options that could provide the low speed, high precision that is needed to balance the SBCA. From the literature review, it was clear that all the other balancing cubes made use of some kind of motor controller running in closed-loop, field-oriented control, and that this was crucial to their workings. However, due to budget constraints, this was simply not possible for the SBCA. The only controller that could meet the budget requirements is the BLHeli-S 30A Drone ESC, or another similar open-loop controller. However, this controller is unable to provide low-speed control, even though it is an ideal option in all other regards.

Experimentation

Fortunately, a 2300 Kv drone motor and a BLHeli-S 30A ESC were available from previous projects, providing a way to test and evaluate their potential and disadvantages. A test rig was assembled, with a rudimentary reaction wheel attached to the motor to add inertia, and the motor was secured down. The motor was wired up to the ESC and an ESP32 to test its performance. The ESP32 was flashed with a simple test program making use of a Bi-Directional DShot library (This library turned out not to work as expected, and details of this are provided further below in subsection 3.6.7). Once the library issues were fixed, the motor was run through a series of tests.

Tests: Firstly, the motor was run at a variety of duty cycles to find the minimum speed at which the motor could still operate smoothly. Next, the motor was run near this duty cycle, and then the duty cycle was stepped up by 0.01 to determine how it would react. This was repeated at different duty cycles. Next, the motor was run in the forward direction, and then the direction was reversed to determine how the controller would handle a change in direction.

The ESC can be programmed with a number of different parameters, such as timing angle and ramp-up factor. These affect how and when the ESC provides power to the motor, and affect its performance. Different combinations of these settings were tested to find the best option.

Findings: The full results of these tests can be found in section 4.1; however, the key takeaways are given below:

- The motor runs stably at duty cycles over 0.05, but is most stable with a duty cycle of 0.075
- The rpm telemetry is unreliable at low rpm
- The motor ‘stalls’ with a sudden direction change, causing it to heat up very fast, and risk damaging the motor.
- The motor’s speed is fairly linear to the duty cycle.
- The motor’s speed is adequately controllable at higher rpm.
- The controller needs to avoid fast changes in direction.

Conclusion: The BLHeli-S 30A ESC demonstrated adequate control precision and responsiveness at high rotational speeds. However, it was unable to maintain stable or accurate control at low rotational speeds. This behaviour aligns with the expected performance characteristics of the ESCs, which are optimised for high-speed operation rather than precise low-speed torque control.

Unless the SBCA can be designed to operate with continuously spooled reaction wheels maintained above a minimum speed threshold, this type of ESC will not be viable for achieving stable balance control. The next step was therefore to investigate whether maintaining the reaction wheels at a moderate constant speed could be possible, as this was not explored in any of the literature reviewed.

Simulation

This next step required both an in-depth working simulation environment and a working controller to simulate the plausibility of using the BLHeli-S 30A ESC. This work was done in conjunction with section 3.8 and section 3.10, the LQR controller design and simulation, covered later in this report. The goal of this simulation was to determine whether it is possible to balance the cube while keeping the motors spinning above a minimum speed. During the controller design (section 3.8), a despool controller was formulated. This controller shifts the cube orientation slightly to slowly despool the motors down to a stationary state. By adding a constant offset to the stationary state, this controller will instead aim to keep the motors spinning at a constant speed. The hope is that maintaining this offset would ensure the motors are always kept spinning, always avoiding the slow-speed region where the ESC loses accurate control over the motor.

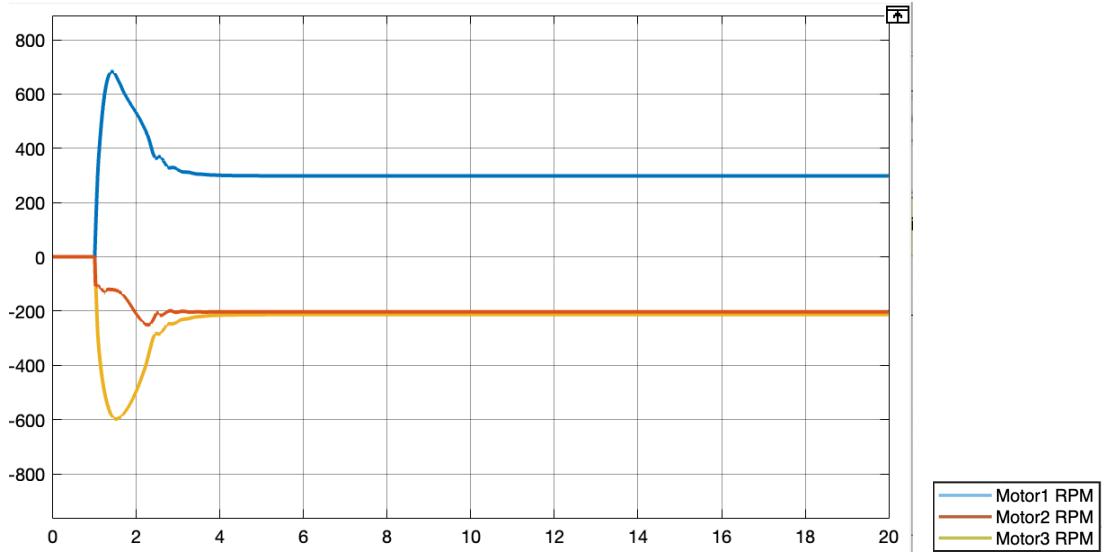


Figure 3.6: SBCA’s three reaction wheel speeds (rad/s) vs time (s) during jump up and balancing on a corner

Figure 3.6 shows that during the jump-up procedure, the three motors all remained spinning at a constant speed. This constant speed can be adjusted in the controller parameters. This constant speed must be high enough to ensure that when the cube experiences a disturbance, it still has enough ‘room to recover’ from the disturbance without bringing the motor to a stop or changing direction.

It was also found that the offsets of the motors need to be applied in the direction in which the motors will need to spin to perform the jump. This ensures the controller does not need to change the direction of the motors to achieve the offset.

To handle changes in direction, a simple motor driver was written, to block changes in direction until the motor is operating below a minimum speed. This is designed to avoid motor stalls, which have the potential of damaging the motors.

Most importantly, this simulation strongly suggests that it will be possible to balance the cube while keeping the motors above a minimum speed.

Decision:

It was therefore decided to go ahead with the BLHeli-S 30A ESC paired with the DRN-2212 motor. This places some limitations and complexities on the design, but it is seen

as the only viable way to build the SBCA while staying in budget.

3.6.3 Cube

Different construction options were considered for the physical structure of the cube. Drawing from the literature, CNC-cut aluminium and carbon fibre were investigated, as these would provide an extremely durable structure, with minimal weight. Unfortunately, both options were quickly ruled out due to their high prices. It was decided that a 3D-printed frame would be the best option, due to its ease of construction and lower cost. This process followed an interactive approach, with the following goals:

- The cube must be as small as possible, while fitting all components inside, including 10 cm diameter reaction wheels
- The cube must prioritise a centre of mass as close as possible to the balancing corner
- The cube must be as light as possible
- The cube must be strong enough to handle the forces that it will experience, up to 0.3 Nm of torque from each motor
- The cube may experience additional forces, mainly vibrations from not-perfectly-balanced reaction wheels

Initial Design

The initial design took inspiration from the self-balancing cubes explored in the literature review. It consists of three reaction wheels placed just inside the cube faces, with their motors in the centre of the cube. The electronics are also placed in the centre of the cube.

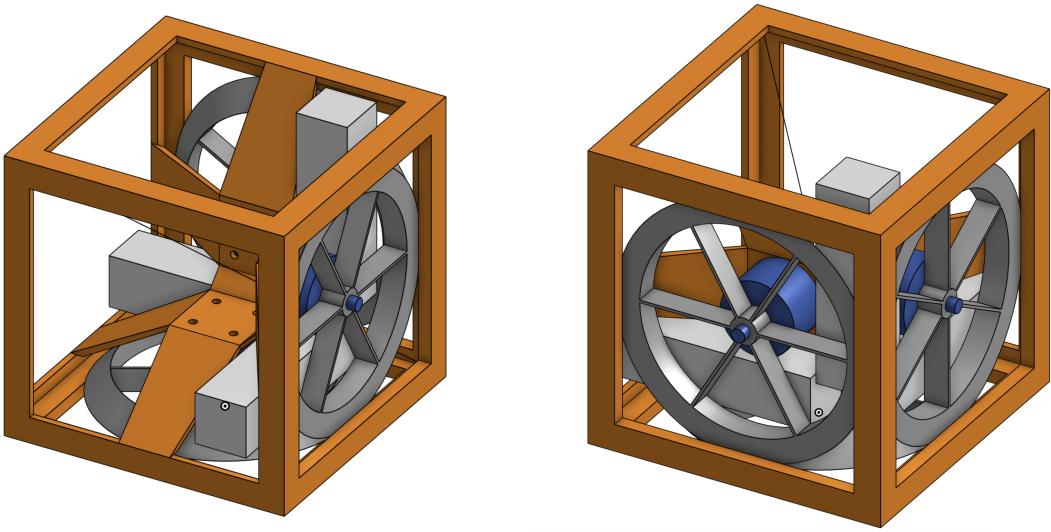


Figure 3.7: CAD views of the first version of the SBCA

Figure 3.7 shows the initial design, with the cube body in orange, motors in blue, reaction wheels in silver, and batteries in grey. Electronics are neglected from this CAD model. Based on the CAD model, this cube had an expected weight of 658 g, with a side length of 110 cm, and a reaction wheel diameter of 96 mm, falling near the desired specifications. This design was deemed to be unsatisfactory for a number of reasons, including concerns about the strength of the central structure that the motors attach to, concerns about how the cube would be assembled, and finally, concerns about how best to 3D print its complex shape. This cube design made use of an initial reaction wheel design, which was later changed (see subsection 3.6.4), further encouraging a complete redesign.

Second Design

The design philosophy for the second design was to move the reaction wheels to the inside of the cube, with the motors attaching to the outside walls of the cube. This was done with the hopes of increasing strength, rigidity, and 3D-printability.

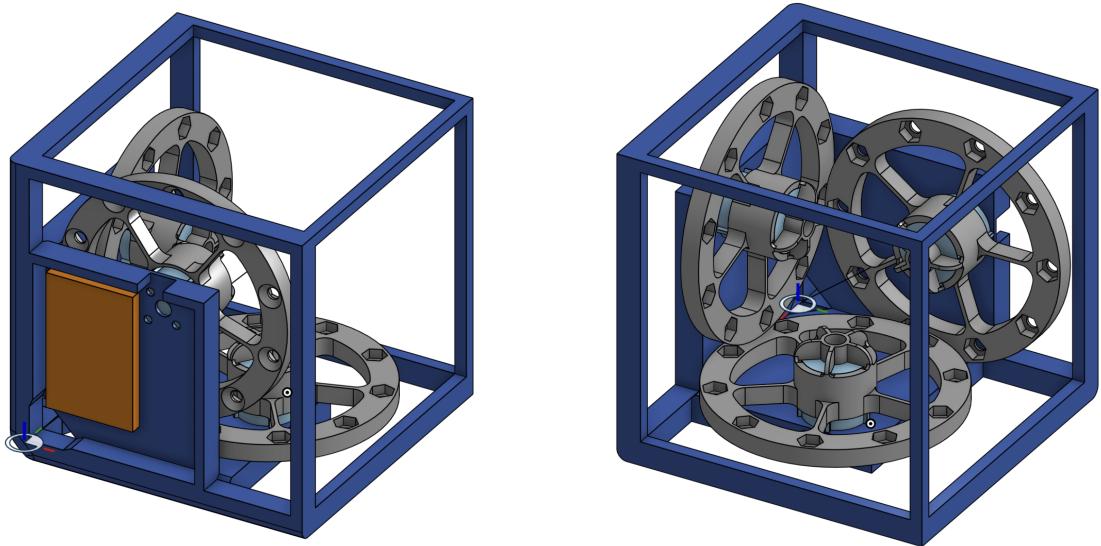


Figure 3.8: CAD views of the second version of the SBCA

Figure 3.8 shows the second version of the SBCA cube design. With the body in blue, batteries in orange, motors in teal, and reaction wheels in grey. This design is based on the final design of the reaction wheels, with the three motor reaction wheel assemblies attached to the outer faces of the cube. The cube is designed to be as light as possible, with extra material only in areas where it is deemed necessary, such as the sections supporting the motor assemblies. This design has an expected weight of 600 g, with the centre of mass shifted 35 mm towards the corner on which the cube is intended to balance. The cube's side length is 124 mm, with 100 mm diameter reaction wheels.

Material

It was initially intended to 3D print the cube structure from PETG material. This was chosen due to its high toughness, low cost, and ease of printing. Following completion of the final design, including simulation in section 3.10, the cube structure was printed and assembled. Unfortunately, upon initial testing, the cube suffered from a ‘rapid unscheduled disassembly’. The cube’s frame shattered and broke apart after two reaction wheels touched, due to the flexibility of the PETG material. The outcome can be seen below:

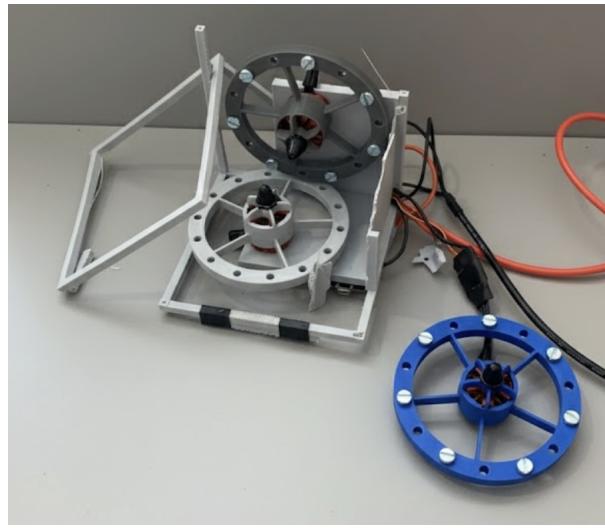


Figure 3.9: Exploded SBCA prototype

Although the design was satisfactory, the strength and rigidity were not. Strengthening the design by adding more material was considered, but this would come at the cost of extra weight, potentially hampering the cube's ability to jump up and balance. So alternative materials were considered, including GF-ABS, CF-ASA, CF-PA12, and finally CF-PPA. After investigating these material and their properties, it was decided to print the cube frame out of CF-PPA, due to its extremely high strength and stiffness.

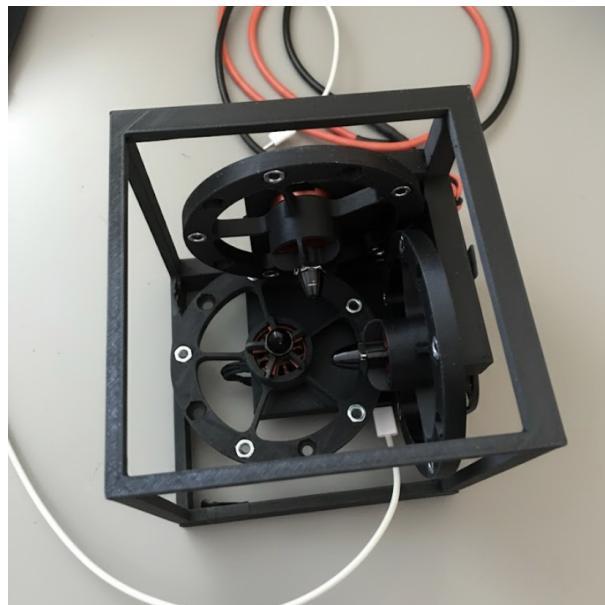


Figure 3.10: Final SBCA Design

3.6.4 Reaction Wheel

Taking inspiration from existing literature, the initial design of the reaction wheels was intended to be CNC machined from aluminium to maximise strength and inertia. However, this approach was later found to be cost-prohibitive within the project's constraints. So, the design approach used by Rem-RC [5] was adopted, which uses 3D-printed reaction wheels with bolts embedded near the rim to increase mass and moment of inertia.

The wheel design process followed an iterative approach similar to that used for the cube's structure and is therefore only briefly discussed here. The initial prototype featured a simple flat wheel mounted above the motor shaft, with protruding bolts used as adjustable masses. Through successive iterations, the design was refined to a compact configuration that encloses the motor, positioning the added mass as close to the cube's outer walls as possible. The final design also incorporates countersunk nuts and bolts to maintain a low profile and reduce the risk of mechanical interference. The reaction wheels were printed out of the same CF-PPA material as the cube structure, with inserted nuts and bolts.

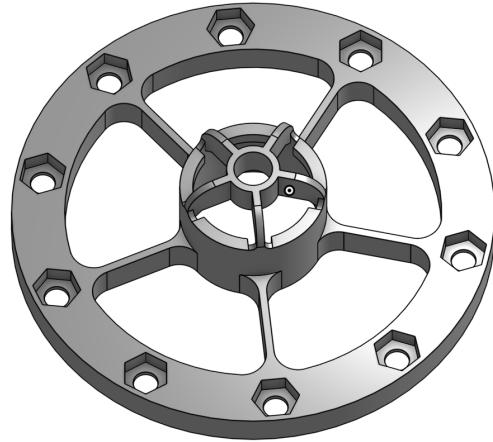


Figure 3.11: Final design of the reaction wheel

3.6.5 IMU

Following the literature review, it was concluded that a single inertial measurement unit (IMU), which includes a gyro and accelerometer, was the best option, due to cost and simplicity. Three different options were looked at.

Option 1: 9 Axis Accelerometer, Gyroscope & Magnetometer IMU: This devboard makes use of an LSM9DS1 9 DOF IMU, with a maximum sample rate of 954 Hz. It provides SPI and I2C interfaces and costs R257.

Option 2: SparkFun 9DoF IMU Breakout: This board makes use of the ICM-20948, another 9 DOF IMU, but with a sample speed of 9 kHz. It costs R268; however, there is limited availability in South Africa.

Option 3: MPU6050 Triple Axis Accelerometer + Gyro 6DOF: This board makes use of the MPU6050, a 6 DOF IMU. It has a sample speed of 1 khz, and is the cheapest at just R52, and is very well supported.

Option 4: DF Robot Fermion: 10 DOF IMU Sensor: This board make use of three independent ICs: the ADXL345 (accelerometer), ITG3205 (gyroscope), VCM5883L (magnetometer), BMP280 (barometer), making it a 10DOF sensor. The gyroscope has a sample rate of 8 Khz. It costs R218.

Decision

It was decided that the greater bandwidth of Options 2 or 4 would help with the performance and stability of the SBCA's controller. Option 4, the DF Robot Fermion: 10 DOF IMU Sensor, was chosen. The DF Robot Fermion: 10 DOF IMU Sensor is fairly old and outdated, lacks ESP32-compatible libraries, and the 4-chip architecture adds extra complexities. Because of this, in retrospect, this may have been the wrong choice, and Option 2 likely was a better choice.

3.6.6 Microprocessor

During the experimentation in section 3.6.2, an ESP32 was used to control the motors. This worked well, and its performance, size and cost were satisfactory. Therefore, it was decided to use it as the SBCA's microprocessor. The following electrical schematic was developed:

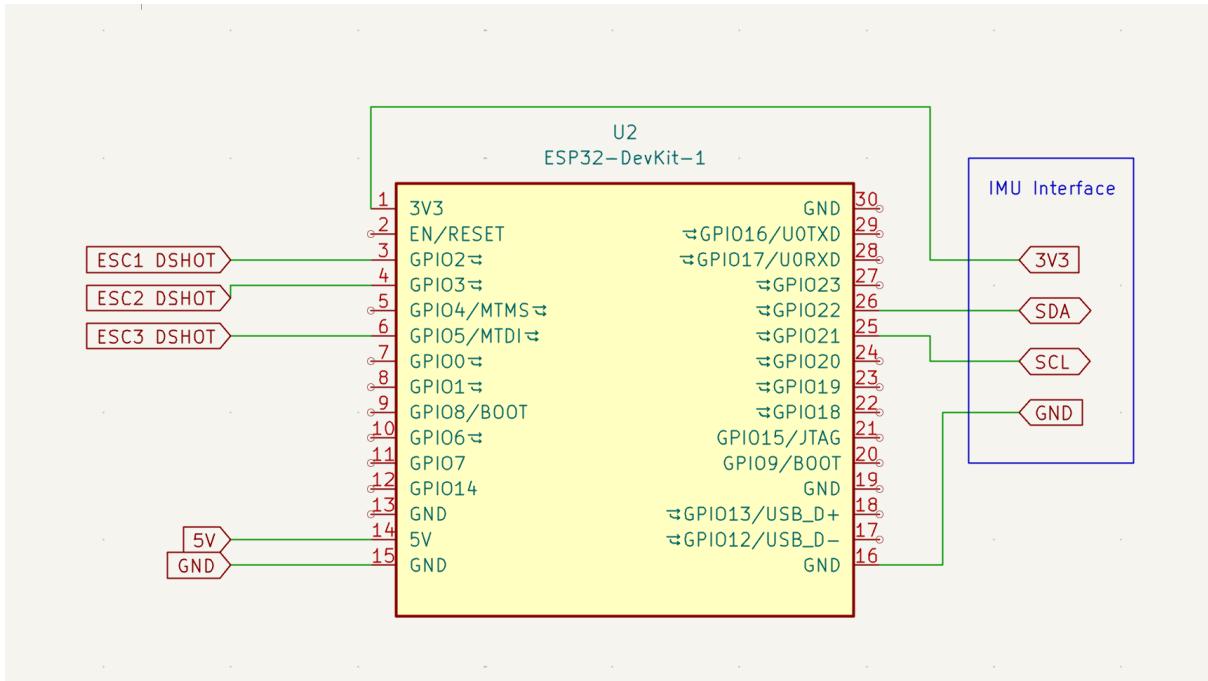


Figure 3.12: SBCA electrical schematic

3.6.7 Library Issues

IMU: DF Robot provides libraries for the selected IMU, the DF Robot Fermion: 10 DOF IMU Sensor; however, upon testing the sensor, a number of unexpected difficulties were encountered. Mainly, the libraries are written for the AVR Arduino board, which has a number of small differences when compared to the ESP32. This includes differences in how the two set up an I2C interface, and the fact that the AVR Arduino Board makes use of 2-byte integer variables, whereas the ESP32 makes use of 4-byte integer variables. After some debugging and troubleshooting, this issue was easy enough to fix, and the library then functioned well.

Bi Directional DShot: The goal is to use the Bi-Directional DShot protocol to receive telemetry data back from the ESC, to provide the controller with the motor speed data. This protocol is not natively supported by the ESP32; however, the DShotRTM GitHub repository provides a well-documented library making use of the ESP32's RTM protocol. Upon testing this library, the ESP32 could successfully send throttle commands to the ESC's however, no telemetry data was received back. A thorough debugging process began. Upon analysing the serial data stream using an oscilloscope, the issue was

determined.

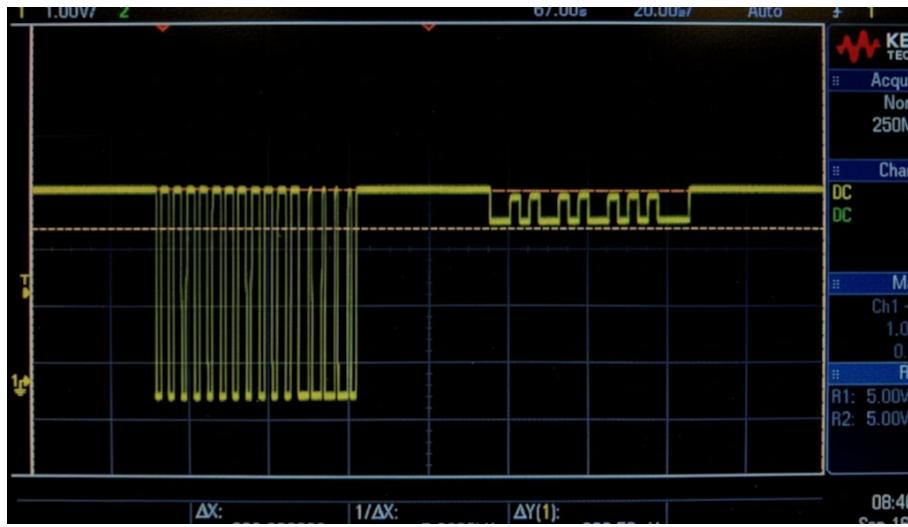


Figure 3.13: DSHOT oscilloscope debugging

In Figure 3.13 above, the throttle command from the ESP32 is seen on the left, with the ESC's RPM telemetry response on the right. It is clear that the ESC is not able to ‘pull down’ the data line adequately, and so the ESP32 does not hear its response. Although this library may work with other ESCs, at least with the ESCs chosen, it does not. This issue was tracked down and determined to be with the RMT peripheral configuration in the library. This issue was opened as a ticket with the developer of the library; however, it remains unsolved. Thankfully, a fork of this original library was found: DShotRMT-NEO. This alternative library functioned well, without the problem described above.

3.6.8 Power Supply

The initial goal of the SBCA was to have it fully self-contained and powered through internal batteries. For this, the SBCA design includes three Li-ion batteries, with one on each face of the cube behind the motors. These three batteries would be wired in series to produce up to 12.6V. Sadly, however, due to time and resource constraints, this was removed from the scope of the project. It was concluded with the project supervisor that this was not a key aspect of the SBCA, and that more time and focus should be spent on the main functionality of the cube. As such, it was decided that the cube will be powered externally through a 12V cable.

3.6.9 Additional Features and Aesthetic Design

The original concept for the Self-Balancing Cube of Awesomeness (SBCA) included several planned ‘awesome’ features, such as integrated LED lighting effects and potential interactive behaviours to enhance its visual appeal. However, due to time and resource constraints, these non-essential features were removed from the project scope in order to prioritise the development and refinement of the core balancing functionality.

Despite the omission of these additional features, the SBCA remains a technically and visually impressive system. Its compact form factor, precise control, and dynamic motion already embody the essence of ‘awesomeness’ that inspired its name.

3.7 Bill of Materials (BOM)

Table 3.7 summarises the main components and materials used in the construction of the Self-Balancing Cube of Awesomeness (SBCA).

Table 3.7: Bill of Materials for the SBCA

Item	Quantity	Unit Cost	Total Cost
M5×8 Countersunk Allen Cap Screws (R69 per 30)	30	2.30	69.00
M5 Nuts (R35 per 20)	30	1.75	52.50
DRN-2212 920KV Brushless Motors	3	220.00	660.00
BLHeli-S 30A ESCs	3	124.00	372.00
ESP32 Development Board	1	102.00	102.00
DFRobot Fermion 10 DOF IMU Sensor	1	218.00	218.00
Cube 3D Print (108 g @ R2000/kg)	1	216.00	216.00
Reaction Wheel 3D Prints (24 g each @ R2000/kg)	3	48.00	144.00
Miscellaneous wiring and connectors	1	50.00	50.00
Total Estimated Cost			R1 963.50

The total cost of approximately **R1 963.50** falls just within the R2 000 project budget. Most of the cost was attributed to the brushless motors, ESCs, and 3D-printed components, while mechanical fasteners and wiring represented a small fraction. Overall, the SBCA design achieved an excellent balance between cost, performance, and system robustness.

Hardware Section Conclusion

The hardware development of the Self-Balancing Cube of Awesomeness (SBCA) followed a structured and iterative process, informed by analytical modelling, simulation, and practical experimentation. Each subsystem, from the motors and drivers to the reaction wheels, sensors, and supporting electronics, was selected and designed to meet the system's functional and performance requirements while remaining within the project's budget constraints.

Although minor compromises were made in component selection due to cost and availability, the final hardware assembly achieved an optimal balance between performance, robustness, and manufacturability. The modular design also allows future improvements, such as refined reaction wheels, integrated lighting, or upgraded motor controllers. Overall, the completed hardware successfully establishes a reliable physical platform for implementing and testing the SBCA's control algorithms.

3.8 LQR Controller Design

The self-balancing cube is an inherently unstable system, requiring active control to maintain a desired orientation. A state-feedback controller based on linear quadratic regulator (LQR) theory with angle-dependent gain blending to handle both small and large orientation errors was implemented.

3.8.1 Rotational Dynamics of the Cube

Let the cube be modelled as a rigid body with a diagonal inertia matrix:

$$I = \text{diag}(I_x, I_y, I_z) = \text{diag}(4.97 \times 10^{-4}, 4.97 \times 10^{-4}, 4.97 \times 10^{-4}) \text{ kgm}^2 \quad (3.1)$$

These values are obtained from the SBCA's CAD model.

Let $\omega = [\omega_x, \omega_y, \omega_z]^\top$ be the angular velocity of the cube in its body frame, and $\tau = [\tau_x, \tau_y, \tau_z]^\top$ the torques applied by the three reaction wheels.

Then rotational dynamics of a rigid body are described by Euler's equations:

$$I\dot{\omega} + \omega \times (I\omega) = \tau \quad (3.2)$$

By assuming small angular velocities of the reaction wheels, the gyroscopic term $\omega \times (I\omega)$ is negligible, the equation is simplified to:

$$I\dot{\omega} = \tau \quad \text{or} \quad \dot{\omega} = I^{-1}\tau \quad (3.3)$$

Let $\theta = [\theta_x, \theta_y, \theta_z]^\top$ represent the small-angle rotation vector from the desired orientation.

The kinematic relation between angular velocity and orientation is:

$$\dot{\theta} = \omega \quad (3.4)$$

Combining the kinematics and dynamics, we define the state vector:

$$x = \begin{bmatrix} \theta \\ \omega \end{bmatrix} \in \mathbb{R}^6 \quad (3.5)$$

Resulting in the linearised state-space model:

$$\dot{x} = \underbrace{\begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}}_A x + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ I^{-1} \end{bmatrix}}_B \tau \quad (3.6)$$

3.8.2 Linear Quadratic Regulator (LQR)

The LQR approach finds a state-feedback control law:

$$\tau = -Kx \quad (3.7)$$

that minimises the quadratic cost function:

$$J = \int_0^\infty (x^\top Qx + \tau^\top R\tau) dt, \quad (3.8)$$

where $Q \in \mathbb{R}^{6 \times 6}$ and $R \in \mathbb{R}^{3 \times 3}$ are symmetric positive definite weighting matrices.

The optimal gain matrix K is obtained by solving the continuous-time algebraic Riccati equation:

$$A^\top P + PA - PBR^{-1}B^\top P + Q = 0 \quad (3.9)$$

and then:

$$K = R^{-1}B^\top P \quad (3.10)$$

However, this was solved using the MATLAB LQR function.

Two sets of gains are designed to handle different magnitudes of orientation error:

- **Large-angle controller:**

$$Q_1 = \text{diag}(2000, 2000, 2000, 100, 100, 100), \quad R_1 = \text{diag}(50, 50, 100), \quad K_1 = \text{lqr}(A, B, Q_1, R_1)$$

- **Small-angle controller:**

$$Q_2 = \text{diag}(400, 400, 400, 2, 2, 2), \quad R_2 = \text{diag}(100, 100, 100), \quad K_2 = \text{lqr}(A, B, Q_2, R_2)$$

These values were determined empirically. K_1 is more aggressive to correct large deviations, and is primarily used for the jump up, while K_2 is less aggressive and is used to keep the SBCA near the desired orientation. LQR is inherently best for linear problems, so this approach helps compensate for the non-linearities when the cube has large angle deviations.

3.8.3 Angle-Dependent Gain Blending

To smoothly transition between K_1 and K_2 , we compute the orientation error using rotation matrices. Let R_{current} be the current rotation and R_{desired} the target rotation. The rotation error is:

$$R_{\text{err}} = R_{\text{desired}}^\top R_{\text{current}} \quad (3.11)$$

The angle of misalignment θ_{err} is obtained from the trace of R_{err} :

$$\cos \theta_{\text{err}} = \frac{\text{trace}(R_{\text{err}}) - 1}{2}, \quad \theta_{\text{err}} = \arccos(\text{clip}(\cos \theta_{\text{err}}, -1, 1)) \quad (3.12)$$

where clipping ensures numerical safety.

A blending factor α is defined as:

$$\alpha = \begin{cases} 0, & \theta_{\text{err}} \leq 10^\circ \\ 1, & \theta_{\text{err}} \geq 25^\circ \\ \frac{\theta_{\text{err}} - 10^\circ}{15^\circ}, & \text{otherwise} \end{cases} \quad (3.13)$$

The final torque command is then:

$$\tau = -(1 - \alpha)K_1x - \alpha K_2x \quad (3.14)$$

3.8.4 State Vector from Rotation Matrices

For larger rotations, the small-angle approximation may not hold. We compute the orientation error vector using the vee-map of a skew-symmetric matrix:

$$x = \begin{bmatrix} \theta_{\text{err}} \\ \omega \end{bmatrix}, \quad \theta_{\text{err}} = \text{vee}\left(\frac{R_{\text{err}} - R_{\text{err}}^\top}{2}\right) \quad (3.15)$$

where the vee-map extracts the 3×1 vector from a skew-symmetric matrix S :

$$\text{vee} \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (3.16)$$

3.8.5 Torque Saturation and Smoothing

To ensure torques are within actuator limits:

$$\tau \leftarrow \text{clip}(\tau, -\tau_{\max}, \tau_{\max}) \quad (3.17)$$

where τ_{\max} depends on motor and reaction wheel capabilities.

A simple low-pass smoothing is applied to avoid abrupt changes:

$$\tau_{\text{smoothed}} = \begin{cases} \alpha_1 \tau + (1 - \alpha_1) \tau_{\text{prev}}, & |\tau| < |\tau_{\text{prev}}| \\ \tau, & \text{otherwise} \end{cases} \quad (3.18)$$

With τ_{prev} updated at each timestep

This design provides a complete torque-feedback control system capable of stabilising the cube across both small and large orientation errors. This process is based on that found in the literature review.

3.9 Despool Tilt Controller

The despool tilt controller is designed to slightly tilt the cube up to a maximum of 10° in the direction of the angular velocity of the reaction wheels. This allows the reaction wheels to despool, avoiding saturation wind-up, and allows the motors to approach the desired rpm.

Inputs and Outputs

The function takes the following inputs:

- $R_{\text{corner}} \in \mathbb{R}^{3 \times 3}$: Rotation matrix representing the cube balanced on a corner (fixed setpoint).
- $\omega \in \mathbb{R}^3$: Angular velocities of the reaction wheels.
- $R_{\text{current}} \in \mathbb{R}^{3 \times 3}$: Current rotation matrix of the cube (body-to-world).

The output is:

- $R_{\text{desired}} \in \mathbb{R}^{3 \times 3}$: Tilted rotation matrix used as the desired setpoint, fed into the LQR controller.

Step 1: Apply bias and smoothing

The angular velocities are first biased by a constant. This is the desired omega for the reaction wheels:

$$\omega_{\text{biased}} = \omega - \begin{bmatrix} -100 \\ 100 \\ -100 \end{bmatrix}$$

A simple exponential smoothing is applied to reduce high-frequency noise:

$$\omega_{\text{smooth}}[k] = (1 - \alpha) \omega_{\text{smooth}}[k - 1] + \alpha \omega_{\text{biased}}[k]$$

where α is a smoothing factor ($\alpha = 0.005$ in this implementation). This ensures gradual adjustment of the tilt.

Step 2: Nonlinear scaling and clamping

The angular velocity is modified to accentuate larger deviations:

$$\omega_{\text{scaled}} = \text{sign}(\omega_{\text{smooth}}) \cdot |\omega_{\text{smooth}}|^2 + \omega_{\text{smooth}}$$

Then, each component is saturated to lie within ± 0.4 radians:

$$\omega_{\text{tilt}} = \max(\min(\omega_{\text{scaled}}, 0.4), -0.4)$$

This ensures that the tilt angle remains small.

Step 3: Compute tilt rotation matrix

The small tilt is applied by converting the angular velocity vector into a rotation matrix using Euler angles:

$$R_{\text{tilt}} = R_z(\text{yaw})R_y(\text{pitch})R_x(\text{roll}),$$

where:

$$R_x(\text{roll}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\text{roll}) & -\sin(\text{roll}) \\ 0 & \sin(\text{roll}) & \cos(\text{roll}) \end{bmatrix}$$

$$R_y(\text{pitch}) = \begin{bmatrix} \cos(\text{pitch}) & 0 & \sin(\text{pitch}) \\ 0 & 1 & 0 \\ -\sin(\text{pitch}) & 0 & \cos(\text{pitch}) \end{bmatrix}$$

$$R_z(\text{yaw}) = \begin{bmatrix} \cos(\text{yaw}) & -\sin(\text{yaw}) & 0 \\ \sin(\text{yaw}) & \cos(\text{yaw}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Here, the input Euler angles are taken as half of the smoothed angular velocity:

$$[\text{roll}, \text{pitch}, \text{yaw}]^\top = \frac{\omega_{\text{tilt}}}{2}$$

Step 4: Apply tilt to corner set point

Finally, the desired rotation matrix is computed by applying the small tilt to the fixed corner set point:

$$R_{\text{desired}} = R_{\text{tilt}} R_{\text{corner}}$$

This produces a slightly tilted desired orientation, allowing the reaction wheels to reduce stored angular momentum while maintaining stability. This method can be repeated for the SBCA balancing on its edge, by simply replacing R_{corner} with R_{edge} .

3.10 Simulation

Building upon the preliminary analysis and initial modelling phase, a full simulation was developed to capture the dynamic behaviour of the self-balancing cube under closed-loop control. This simulation included the complete control system, including both the state-feedback LQR controller and the despool tilt controller, providing an accurate representation of the system's physical response.

Similar to the initial simulation (section 3.5), this simulation was made using MATLAB and Simulink, using the Multibody toolkit to model the cube's rigid-body dynamics. This

environment provides a physically realistic multibody representation without the need for explicit derivation of the coupled nonlinear equations of motion.

3.10.1 Model Setup

The cube was modelled as a solid brick. The parameters for mass, inertia, and geometry were directly imported from the final CAD assembly.

Three internal reaction wheels were modelled as rigid disks mounted orthogonally within the cube using ‘Revolute Joint’ constraints. The model parameters were imported from the final CAD design.

3.10.2 Controller Implementation

The LQR controller was implemented in a MATLAB Function block and received the cube’s orientation and angular velocity vectors as inputs. The controller generated the desired torque commands for each reaction wheel, which were subsequently converted to motor commands.

In addition, the despool tilt controller described in section 3.9 was integrated in series with the LQR controller. This module applied a smooth, bounded shift to the corner setpoint, biasing the cube’s orientation by a small angle. This gradual tilting action allows the reaction wheels to despool accumulated momentum.

3.10.3 Motor and Actuator Modelling

Each reaction wheel was driven by a simulated brushless DC motor governed by an electronic speed controller (ESC). The motor dynamics were modelled through a custom MATLAB Function block implementing a nonlinear relationship between duty cycle input, back-EMF, and torque output. The motor model is given by:

$$\tau_{\text{motor}} = \text{ESC_Motor}(d, \omega_w) \quad (3.19)$$

where d is the duty cycle command and ω_w is the current angular velocity of the wheel.

3.10. SIMULATION

The function computes the torque based on the following relationships:

$$V_{\text{batt}} = 3.7 \times 3 = 11.1 \text{ V}, \quad (3.20)$$

$$K_t = \frac{60}{2\pi k_V}, \quad K_e = K_t, \quad (3.21)$$

$$I = \frac{V_m - K_e \omega_w}{R}, \quad (3.22)$$

$$\tau_e = K_t I, \quad (3.23)$$

$$\tau_{\text{motor}} = \tau_e - \tau_{\text{loss}} \quad (3.24)$$

where $Kv = 920 \text{ RPM/V}$ is the motor velocity constant, $R = 0.3 \Omega$ is the phase resistance, and τ_{loss} includes both viscous and aerodynamic drag components modelled as:

$$\tau_{\text{loss}} = c_v \omega_w + c_d \omega_w^2 \quad (3.25)$$

with $c_v = 10^{-4}$ and $c_d = 10^{-6}$ representing empirically chosen coefficients. The ESC mapping incorporates a deadband near zero input, ensuring realistic throttle response.

The accuracy of this model will be explored in section 4.2.

3.10.4 Simulation Objectives

The full simulation aims to validate the closed-loop performance of the system under realistic dynamic and electrical constraints. This simulation, therefore, bridges the gap between theoretical control design and experimental implementation, providing a high-fidelity virtual environment to evaluate the SBCA's controller's performance.

Chapter 4

Results

4.1 Motor Results

The motor tests described in subsection 3.6.1 were conducted to evaluate the performance of the drone motors and ESCs, specifically to determine whether they provided acceptable control and to analyse their low-speed characteristics. During the tests, data was logged comparing the ESC-reported motor rpm with the applied duty cycle percentage. The results have been plotted after scaling the maximum rpm to correspond with the maximum duty cycle, allowing for a clearer visualisation of the linear relationship between motor speed and duty cycle.

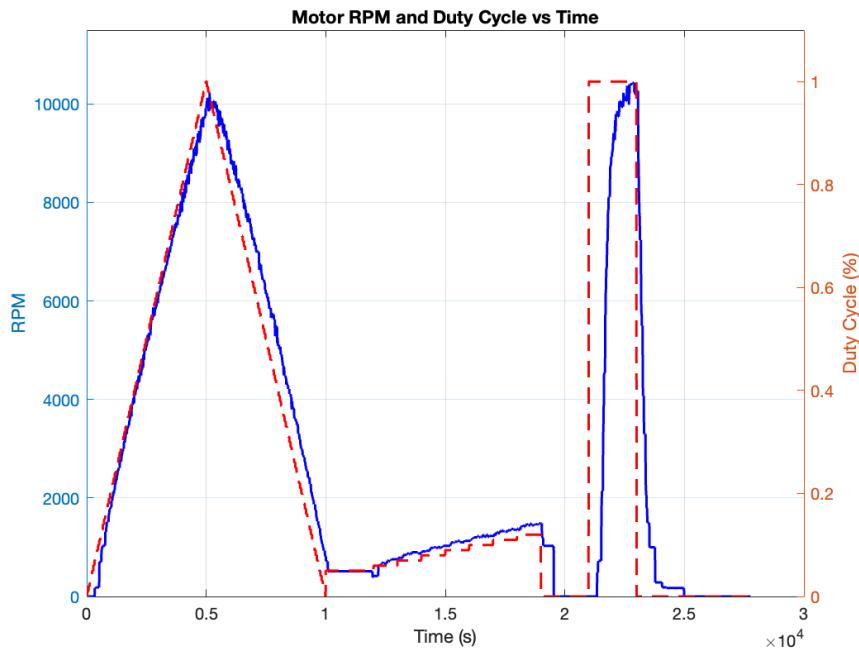


Figure 4.1: Motor rpm and duty cycle, vs time during full test

Figure 4.1 shows the logged data from the full test. The main takeaway from this test was the linear relationship between the duty cycle set point and the corresponding motor

4.1. MOTOR RESULTS

RPM. This data will also be used later, in section 4.2, to compare the simulated motor and ESC.

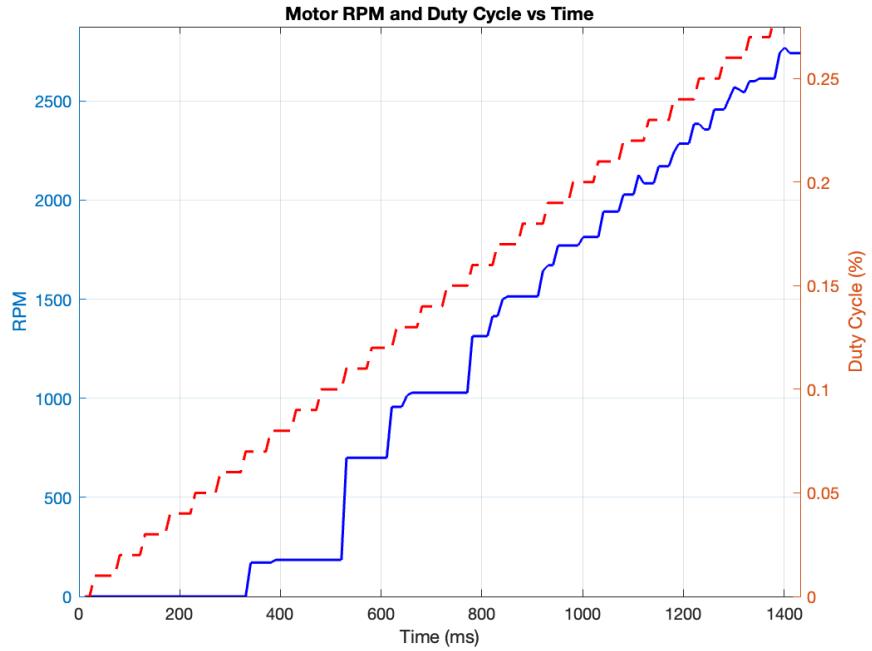


Figure 4.2: Motor rpm and duty cycle, vs time during ramp up

Figure 4.2 presents a close-up view of the motor's reported rpm during the initial phase of the ramp-up. The data indicate that the ESC reports the motor as spinning only once the duty cycle reaches approximately 0.075. However, the reported rpm may be unreliable at low speeds, as shown by the pronounced stepping observed in the rpm curve. This inaccuracy is further confirmed by direct observation of the motor, which began to spin stably at a duty cycle closer to 0.05. Below a duty cycle of 0.05, the motor jumps erratically.

4.1. MOTOR RESULTS

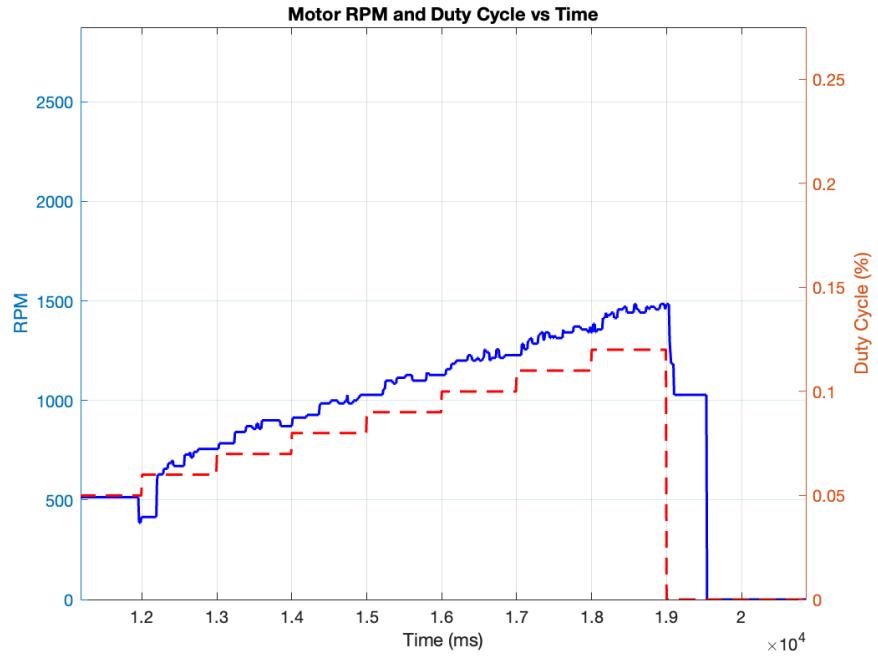


Figure 4.3: Motor rpm and duty cycle, vs time during ramped step

Figure 4.3 was an attempt to see how the motor would respond to a series of small steps, and if it would provide enough accuracy. The figure shows that the motor can successfully track the setpoint; however, it struggles to resolve individual steps.

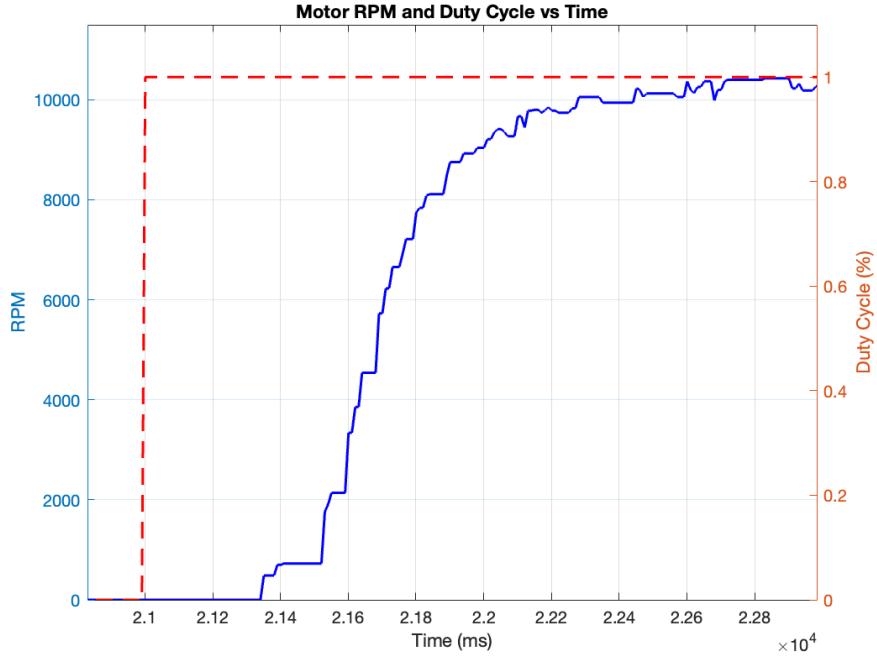


Figure 4.4: Motor rpm and duty cycle, vs time during full step

4.1. MOTOR RESULTS

Figure 4.4 shows the motor's response to a step from a duty cycle of 0 to 1. The motor takes approximately 341 ms before it starts to report any rpm, and then reaches 63% of its final value in approximately 380 ms.

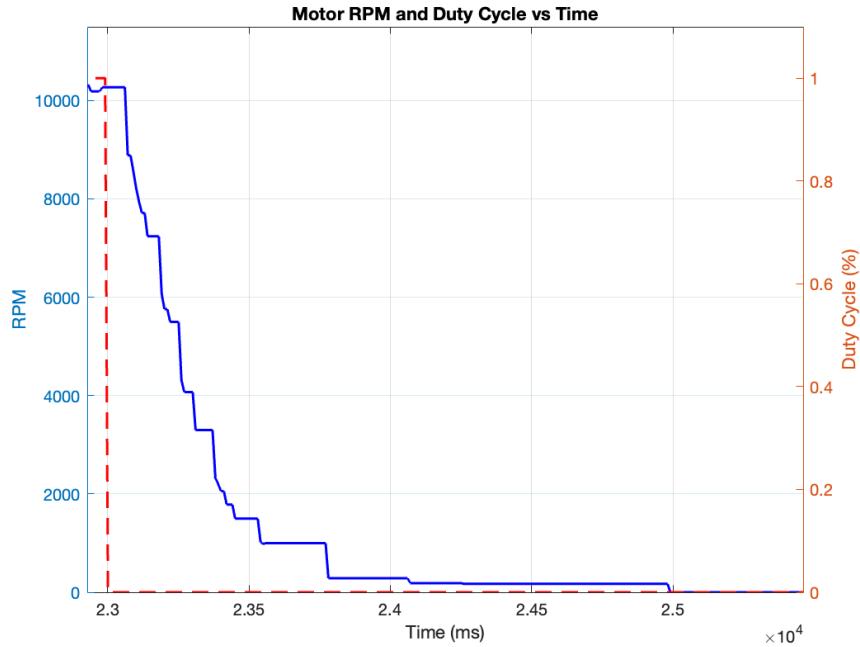


Figure 4.5: Motor rpm and duty cycle, vs time during full step down

Figure 4.5 shows the motor's response to a step from a duty cycle of 1 to 0. The motor takes approximately 61 ms before it starts to report any rpm change, and then reaches 63% of its final value in approximately 260ms. This is an interesting result, showing that the motor responds better when slowing down.

The final test was to step the motor from the positive direction to the negative direction, to see how it would handle this crossover. Upon the first test, the motor stalled, made a rough sound, and quickly began to overheat before emitting smoke. Fortunately, once it was left to cool down, it continued to function correctly. This needs to be carefully avoided in the final SBCA.

4.2 Simulation Results

4.2.1 Motor Simulation

The full duty cycle sweep from section 4.1 was imported into the Matlab Simulink model. This duty cycle sweep was applied to one of the simulated reaction wheels, and the resulting RPM was recorded.

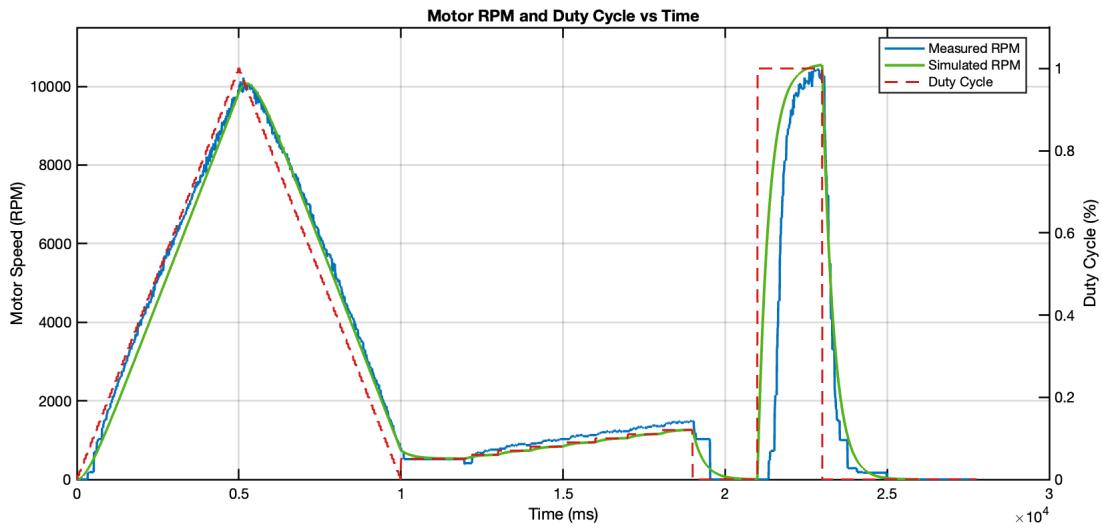


Figure 4.6: Comparison between simulated and physical motor response

Figure 4.6 shows that the simulated motor response closely follows the experimentally measured response. The primary difference occurs during the step-up phase, where the physical motor shows a noticeable delay compared to the modelled response. This lag is likely caused by the motor struggling to start up and momentarily stalling. Once the motor begins spinning, its behaviour aligns closely with the simulated curve. Overall, this validates our simulation model.

4.2.2 Edge Jump, and Balance Simulation

The full simulation was set up, with the desired set point of the cube balancing on its edge. The simulation was run, and the following was recorded:

4.2. SIMULATION RESULTS

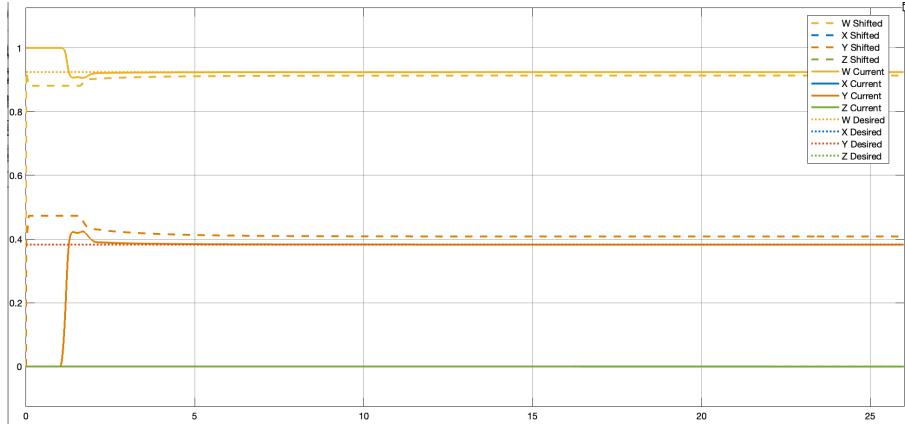


Figure 4.7: Edge jump quaternions over time (s)

Figure 4.7 presents the simulated quaternion components over time during the edge jump-up procedure. The plot displays three trajectories: the desired edge orientation (dotted line), the despool-shifted orientation (dashed line), and the cube's actual orientation (solid line). The results clearly illustrate how the despool controller gradually adjusted the setpoint to offload accumulated wheel momentum following the jump-up event. The LQR controller was shown to effectively track the time-varying setpoint with satisfactory accuracy, maintaining stable attitude control throughout the despool and balancing phase.

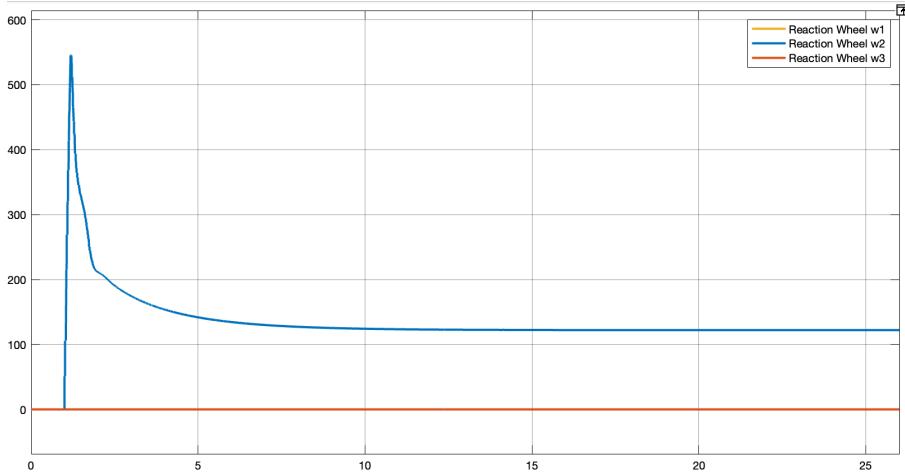


Figure 4.8: Reaction wheel speeds (rad/s) vs time (s) during edge jump

Figure 4.8 presents the reaction wheel angular velocities during the edge jump operation. The wheels remained within their operational limits, avoiding saturation, and then subsequently despooled gradually to the desired speeds defined by the despool controller. This behaviour confirms the design premise of the SBCA, demonstrating that the chosen combination

4.2. SIMULATION RESULTS

of motors, reaction wheels, and system mass is viable. The results also validate the effectiveness of the despool controller, showing that it performs as intended within the simulation environment.

4.2.3 Corner Jump, and Balance Simulation

The full simulation was set up, with the desired set point of the cube balancing on its corner. The simulation was run, and the following was recorded:

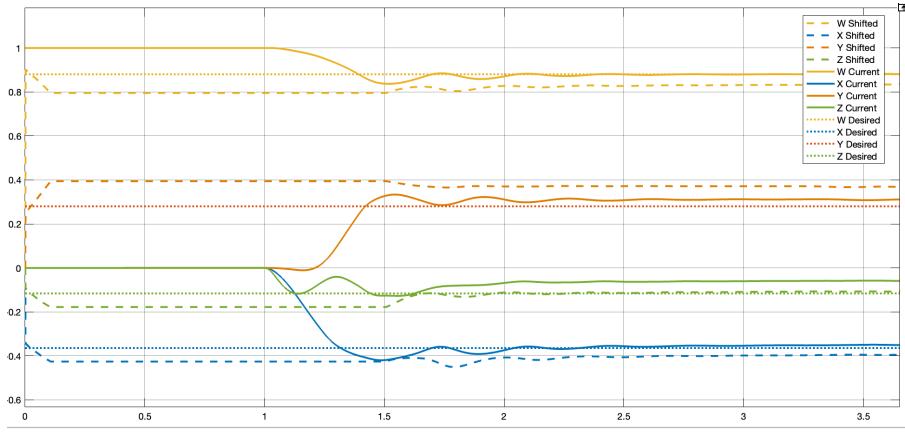


Figure 4.9: Corner jump quaternions over time

Figure 4.9 presents the simulated quaternion components over time during the corner jump-up procedure. The plot displays three trajectories: the desired corner orientation (dotted line), the despool-shifted orientation (dashed line), and the cube's actual orientation (solid line). The results clearly illustrate how the despool controller gradually adjusts the setpoint to offload accumulated wheel momentum following the jump-up event. The LQR controller was shown to effectively track this time-varying setpoint with satisfactory accuracy, maintaining stable attitude control throughout the despool and balancing phases.

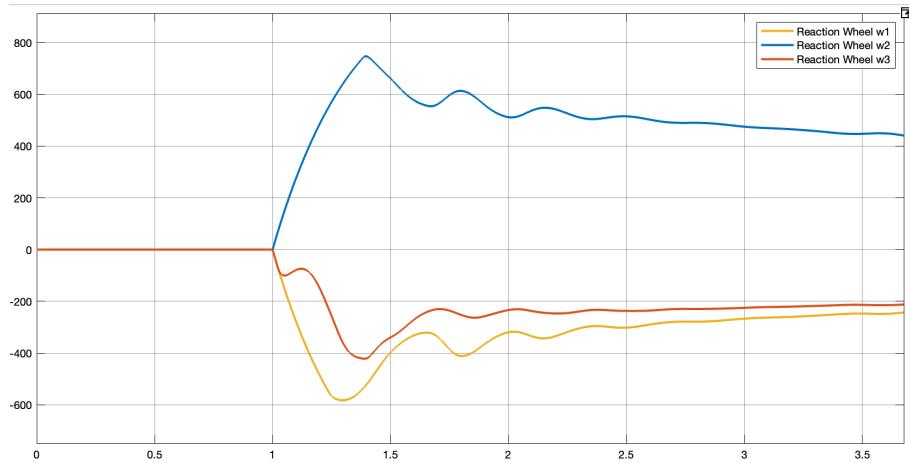


Figure 4.10: Reaction wheel speeds (rad/s) vs time (s) during corner jump

Figure 4.10 presents the reaction wheel angular velocities during the corner jump operation. The wheels remain within their operational limits, avoiding saturation, and despool gradually to the desired speeds defined by the despool controller. This behaviour confirms the design premise of the SBCA, demonstrating that the chosen combination of motors, reaction wheels, and system mass is viable for the corner setpoint. Furthermore, the results validate the effectiveness of the despool controller, showing that it performs as intended within the simulation environment.

4.2.4 Simulation Results Conclusion

From these results, it was concluded that the simulation provides strong evidence that the SBCA will perform as intended under real-world conditions. The system successfully demonstrates stable corner balancing, effective despooling of the reaction wheels, and accurate attitude tracking through the combined action of the LQR and despool controllers. The agreement between the simulated dynamics and the theoretical design expectations reinforces the validity of the modelling approach and control architecture.

The simulation confirms that the selected hardware configuration, including the motor torque capacity, reaction wheel inertia, and overall mass distribution, is sufficient to achieve both stability and responsiveness. While minor discrepancies between the model and a physical prototype are expected, the results strongly indicate that the core design principles are sound.

4.3 Balancing Results - Physical Testing

Following the construction and programming of the final Self-Balancing Cube of Awesomeness (SBCA) prototype, a series of experiments were conducted to evaluate its balancing performance.

The results presented in this section demonstrate the cube's ability to balance itself, beginning with the edge-balancing configuration.

4.3.1 Edge Balancing

In this test, the SBCA was programmed to balance on one of its edges using the implemented control algorithm. The cube successfully jumped up to the edge and balanced there. The following data was recorded.

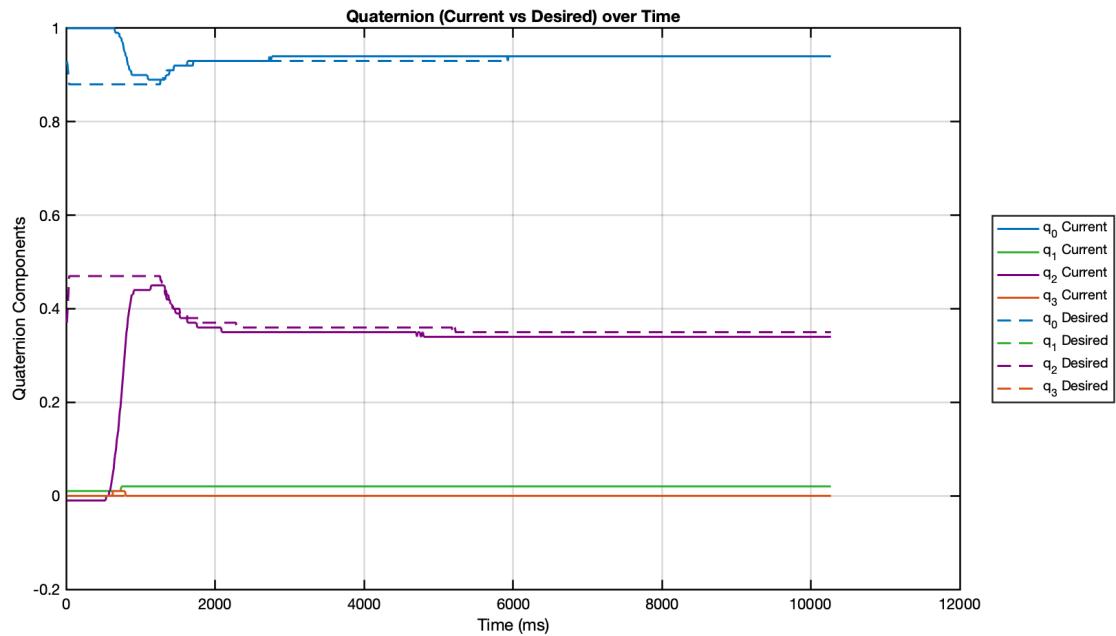


Figure 4.11: Edge jump quaternions over time during edge jump

Figure 4.11 illustrates the quaternion components of the cube's current and desired orientations over time.

From these results, it is evident that the SBCA successfully maintains a stable balance

4.3. BALANCING RESULTS - PHYSICAL TESTING

on its edge. This behaviour was also confirmed through visual observation during testing. The system demonstrated satisfactory responsiveness and stability, with minimal oscillations once equilibrium was achieved. This data also closely resembles the simulated data.

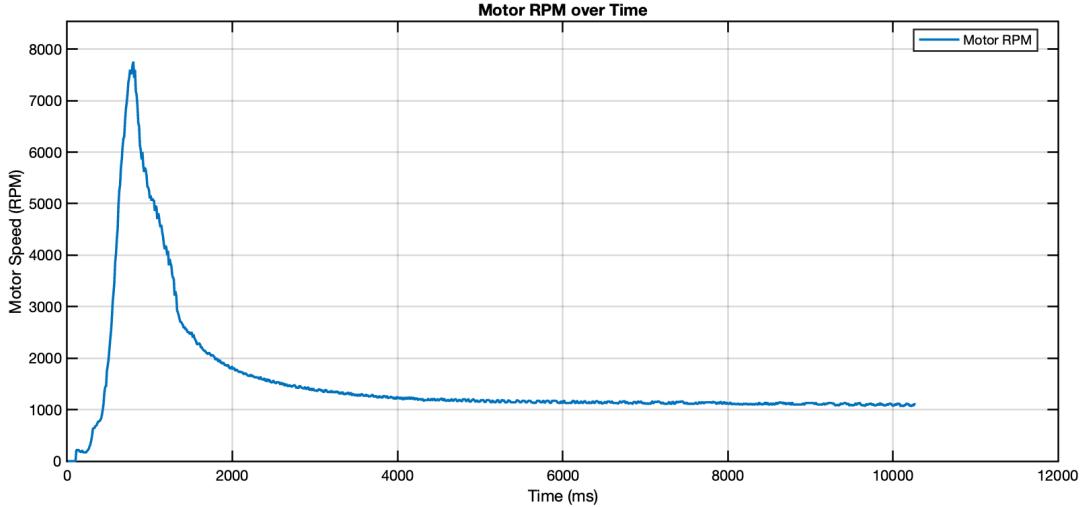


Figure 4.12: Reaction wheel speed (rpm) during edge jump

Figure 4.12 illustrates the reaction wheel speeds (in RPM) during the edge jump manoeuvre. From these results, it is evident that the despool controller successfully reduced the wheel speed to the desired value. The measured behaviour closely resembled the simulated data, confirming that the implemented controller performed as expected in the physical system.

Finally, small disturbances were applied to the cube, and the SBCA was observed to be able to quickly recover from these.

4.3.2 Corner Balancing

In this test, the SBCA was programmed to perform a jump-up manoeuvre from rest and balance on one of its corners using the implemented control algorithm. The cube successfully executed the jump, transitioned to the corner position, and maintained a stable balance. The following results were recorded during the test.

4.3. BALANCING RESULTS - PHYSICAL TESTING

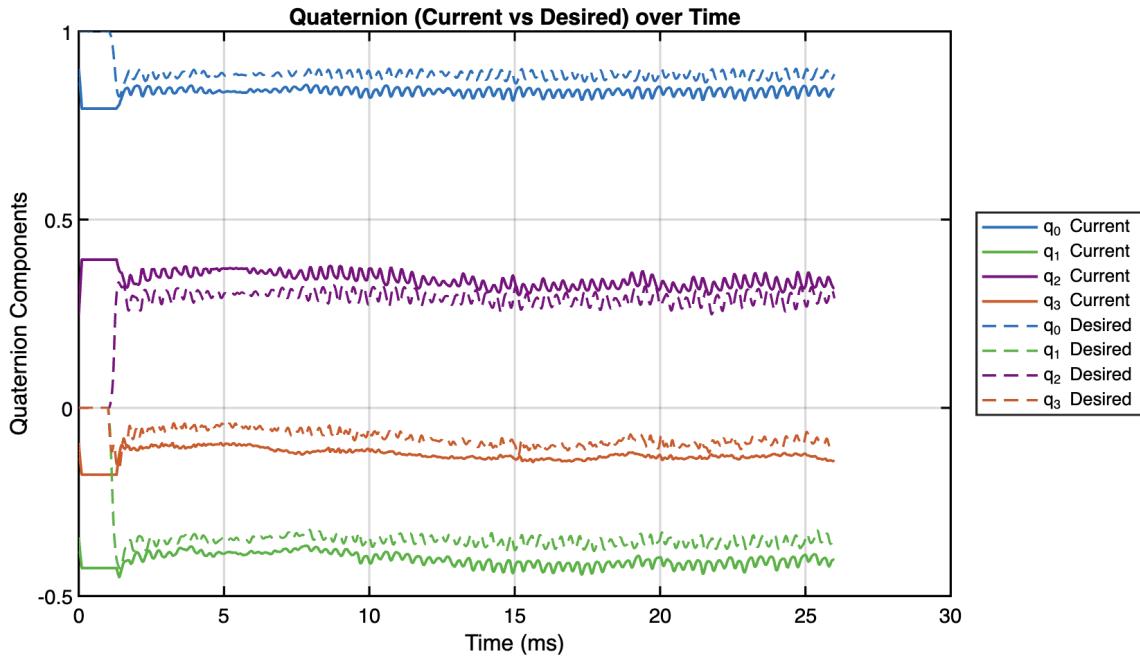


Figure 4.13: Corner jump quaternions over time during corner jump

Figure 4.13 shows the quaternion components of the cube's current and desired orientations over time. From these results, it is evident that the SBCA successfully achieved and maintained a stable balance on its corner. This behaviour was further confirmed through visual observation during testing. The system exhibited good responsiveness and stability; however, it did have small oscillations once equilibrium was reached.

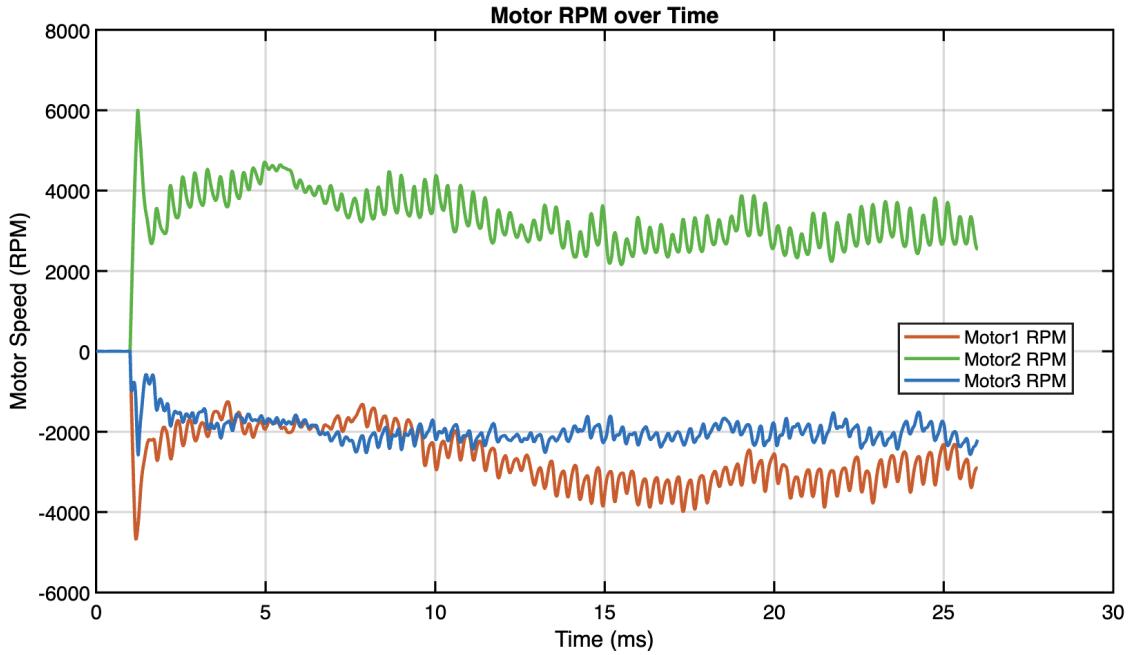


Figure 4.14: Reaction wheel speeds (rpm) during Corner Jump

Figure 4.14 shows the reaction wheel speeds during the corner jump manoeuvre. It shows that the despool controller effectively reduced the wheel speed to the target value. This ensures that the motors do not saturate, as well as ensuring they remain above minimum speeds, and do not stall or lose controllability. The measured response closely followed the simulation results, confirming that the implemented control strategy performed as expected in the physical system.

Finally, small disturbances were applied to the cube, and the SBCA was observed to be able to quickly recover from these.

4.4 Results Conclusion

The experimental results demonstrate that the Self-Balancing Cube of Awesomeness successfully achieved all primary design objectives and functional requirements. The system was able to perform both edge and corner balancing, as well as execute the jump-up manoeuvres, closely matching the behaviour predicted by analytical models and simulations.

4.4. RESULTS CONCLUSION

The implemented control algorithms provided stable and responsive performance, effectively maintaining balance with minimal oscillations and quick recovery from small disturbances. Hardware subsystems, including the reaction wheels, despool controller, and motor drivers, operated reliably and within the design specifications.

Overall, the SBCA met or exceeded the established performance targets, validating the design approach, control strategy, and hardware implementation. The close correlation between simulated and experimental data further confirms that the developed simulation accurately represents the physical dynamics of the cube. In conclusion, the SBCA successfully demonstrates a stable, self-contained, and functional self-balancing cube—living up to its name in both capability and ‘awesomeness’.

Chapter 5

Discussion

The results obtained from both simulation and experimental testing confirm that the Self-Balancing Cube of Awesomeness (SBCA) behaves consistently with the theoretical models and control concepts established in the literature.

The system's ability to perform edge and corner balancing demonstrates that the underlying control strategy, a combination of linear quadratic regulation (LQR) and despool control, provides stable, robust, and responsive performance for an inherently unstable system.

Relation to Theoretical Work

The theoretical foundation of this project, as outlined in the literature review, was drawn primarily from prior work on the Cubli developed at ETH Zürich [4, 3], and similar reaction wheel stabilised systems. These studies established that dynamic balance can be achieved by the transfer of angular momentum through reaction wheels using linear feedback control. The SBCA replicates this fundamental principle, demonstrating that an appropriately tuned LQR controller can stabilise a rigid cube on both its edge and corner despite significant nonlinearities in its dynamics.

Furthermore, the despool control method developed for this project builds upon ideas presented in the Cubli's energy management and jump-up control strategy[3]. By regulating the stored angular momentum in the reaction wheels before and after a jump manoeuvre, the SBCA successfully prevents motor saturation and maintains controllability throughout the manoeuvre. The experimental results show that this approach effectively bridges the gap between the simplified linear model used for controller design and the nonlinear behaviour of the real system, confirming the validity of the analytical assumptions made in earlier sections.

Relevance and Comparison to Existing Work

When compared with prior research, the SBCA achieves similar functional outcomes, including corner balancing and jump-up manoeuvres, but at a significantly lower hardware and implementation cost. For example, where the Cubli utilised precision Maxon EC-45 Flat motors, high-resolution encoders, and industrial-grade drivers, the SBCA accomplished comparable stability using low-cost drone motors, BLHeli-S controllers, and an ESP32 microcontroller. This highlights an important contribution of this work. Advanced control principles such as LQR and momentum management can be implemented successfully using affordable, readily available components without sacrificing key system performance. The ability to reproduce the key behaviours of the Cubli on a limited budget broadens accessibility to research and education in dynamic balancing and nonlinear control.

Interpretation of Results

The experimental data closely matched the simulated behaviour across all tests. The quaternion responses confirmed that the system reached and maintained the desired orientation with minimal overshoot and steady-state error, while the reaction wheel speeds demonstrated the correct despooling dynamics predicted by simulation. These correlations validate not only the controller design but also the accuracy of the system model used in simulation. Minor deviations observed in the physical system, such as low-speed oscillations and small delays in wheel response, can be attributed to sensor noise, mechanical imbalance, and the limitations of the ESC control at low rotational speeds.

Significance of the Results

The SBCA contributes to the growing body of work on small-scale, reaction-wheel-stabilised systems by demonstrating that control performance comparable to established research prototypes can be achieved under strict cost and design constraints. This reinforces the theoretical understanding that attitude control and dynamic balancing can be effectively realised through momentum exchange mechanisms governed by linear control laws, even in systems that are highly nonlinear in practice.

Chapter 6

Recommendations

While the Self-Balancing Cube of Awesomeness (SBCA) achieved all of its primary objectives, several opportunities for improvement and further development have been identified. These recommendations aim to enhance the performance, robustness, and overall functionality of the system, as well as extend its use as an educational and research platform.

Motor Control and Actuation

A key limitation observed during testing was the restricted low-speed torque control of the brushless DC motors driven by BLHeli-S ESCs. These controllers are designed primarily for high-speed drone applications and therefore exhibit poor torque linearity near zero speed. Future iterations of the SBCA would benefit significantly from the implementation of field-oriented control (FOC). This could be achieved by replacing the existing ESCs with dedicated FOC-capable drivers, such as the ODrive or SimpleFOC-based controllers. FOC would allow precise control of motor currents, enabling smoother, more stable balancing performance, faster disturbance rejection, and improved efficiency.

Sensing and State Estimation

The current implementation uses a DFRobot Fermion 10-DOF IMU, which provides adequate performance but is limited by sensor noise and bias drift, and is outdated. For improved control accuracy, it is recommended to adopt a more modern IMU. Additionally, implementing more advanced sensor fusion algorithms could further improve attitude estimation.

Control Strategies

Although the linear quadratic regulator (LQR) provided excellent performance for the linearised system, further work could explore more advanced or adaptive control strategies. Nonlinear control methods could be investigated to enhance robustness across a wider range of operating conditions.

Mechanical and Aesthetic Enhancements

From a mechanical perspective, future prototypes could benefit from the use of stronger, lighter materials such as carbon fibre or aluminium for the cube frame to reduce weight while maintaining rigidity. Furthermore, as part of the original project vision, the addition of aesthetic and interactive features such as LED lighting would enhance the cube's visual appeal. For instance, lighting could be programmed to respond to balance states or control activity, providing intuitive feedback during demonstrations.

Educational and Research Extensions

Finally, the SBCA offers an excellent platform for further research into advanced embedded control and actuation systems. Future iterations could include modular hardware that allows easy swapping of sensors or controllers for comparative studies, or serve as a benchmark platform for teaching modern control techniques such as FOC, or nonlinear control feedback. This would transform the SBCA from a proof-of-concept prototype into a flexible, open-source educational and experimental tool for advanced control system design.

Summary

In summary, future improvements should focus on the integration of FOC-based motor control, improved sensing and estimation, and more sophisticated control algorithms.

Chapter 7

Conclusions

The results presented in chapter 4 demonstrate that the Self-Balancing Cube of Awesomeness (SBCA) successfully meets the primary objectives of this study, namely, to design, model, and implement a robust control system capable of stabilising a cube on one of its edges or corners. Simulation and experimental results confirm that the system behaves in close agreement with theoretical predictions, thereby validating the analytical models and control strategies developed earlier.

From the simulations, it was observed that the LQR controller combined with the despool controller provided smooth, stable, and responsive attitude control. The simulated cube was able to perform both edge and corner balancing manoeuvres without saturating the reaction wheels. The despool algorithm effectively managed stored angular momentum in the wheels, ensuring continued controllability and maintaining system stability.

The experimental implementation further reinforced these findings. The physical SBCA prototype achieved self-balancing on both its edge and corner configurations, exhibiting minimal oscillations and fast recovery following small disturbances. The close correlation between measured and simulated reaction wheel speeds demonstrates that the simulation model accurately represents the real-world system, with only minor discrepancies. These differences, however, did not significantly impact performance, confirming that the chosen hardware and control design were appropriate.

A particularly noteworthy outcome of this project is the achievement of comparable performance to similar systems presented in the literature, but at a fraction of the cost. For example, the original Cubli prototype[4, 3] from ETH Zürich utilised high-end precision motors costing roughly R3500 each. Other projects examined in the literature review also utilised similarly expensive hardware. To contrast this, the SBCA achieves equivalent core functionality with an estimated budget below R2000. This was made possible through careful component selection, creative mechanical design, and the use of affordable yet capable hardware such as the DRN-2212 brushless motors, BLHeli-S ESCs, and an ESP32-based control platform. This demonstrates that high-performance

balance control and dynamic actuation can be achieved on a constrained budget without sacrificing essential functionality or control fidelity.

In addition to validating theoretical models, the SBCA provides a strong experimental foundation for further exploration in multi-axis reaction-wheel control and nonlinear balance dynamics. The project effectively demonstrates how classical control theory, particularly linear quadratic regulation, can be extended to complex, multi-axis systems with practical hardware constraints. The iterative design process, from simulation to hardware prototyping to experimental verification, ensured that the final system was both functional and robust within its design envelope.

While time and resource constraints prevented the inclusion of aesthetic enhancements, such as integrated lighting, these omissions did not detract from the project's technical success. The SBCA is an impressive and fully operational proof-of-concept, serving both as a demonstration platform and as a learning tool for advanced feedback control and embedded mechatronic integration.

Overall, the project bridges the gap between theoretical control design and practical implementation, contributing to the growing body of work on small-scale, dynamically stable robotic systems. The outcomes confirm that with well-structured modelling, thoughtful hardware integration, and systematic controller design, highly nonlinear systems such as the self-balancing cube can be realised and controlled with precision.

Bibliography

- [1] T. Liao, S.-J. Chen, C. C. Chiu, and J. Yan, “Nonlinear dynamics and control of a cube robot,” *Mathematics*, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:225112421>
- [2] R. Piotrowski and A. Kowalczyk, “Optimal state feedback controller for balancing cube,” *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 16, no. 3, p. 3–12, Aug. 2023. [Online]. Available: <https://www.jamris.org/index.php/JAMRIS/article/view/784>
- [3] M. Gajamohan, M. Merz, I. Thommen, and R. D’Andrea, “The cubli: A cube that can jump up and balance,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3722–3727.
- [4] M. Gajamohan, M. Muehlebach, T. Widmer, and R. D’Andrea, “The cubli: A reaction wheel based 3d inverted pendulum,” in *2013 European Control Conference (ECC)*, 2013, pp. 268–274.
- [5] Remrc, “Remrc/self-balancing-cube.” [Online]. Available: <https://github.com/remrc/Self-Balancing-Cube>
- [6] “Nikolatoy®esp32 self balancing cube robot.” [Online]. Available: https://nikolatoy.com/products/nikolatoy%C2%AEesp32-self-balancing-cube-robot?srsltid=AfmBOoorM2jiTENk9JtEzroOvtT_5sXkS29SJl5UaTEnvZJs4XdRm7L
- [7] Z. Chen, X. gang Ruan, and L.-P. Yuan, “Dynamic modeling of a cubical robot balancing on its corner,” 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:55991829>
- [8] S. Trimpe and R. D’Andrea, “The balancing cube: A dynamic sculpture as test bed for distributed estimation and control,” *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 48–75, 2012.
- [9] M. Hofer, M. Muehlebach, and R. D’Andrea, “The one-wheel cubli: A 3d inverted pendulum that can balance with a single reaction wheel,” *Mechatronics*, vol. 91, p. 102965, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957415823000211>

BIBLIOGRAPHY

- [10] F. Bobrow, B. A. Angelico, F. P. R. Martins, and P. S. P. da Silva, “The cubli: Modeling and nonlinear attitude control utilizing quaternions,” *IEEE Access*, vol. 9, pp. 122 425–122 442, 2021.
- [11] R. Mahony, T. Hamel, and J.-M. Pflimlin, “Complementary filter design on the special orthogonal group $so(3)$,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 1477–1484.
- [12] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, “Estimation of imu and marg orientation using a gradient descent algorithm,” in *2011 IEEE International Conference on Rehabilitation Robotics*, 2011, pp. 1–7.
- [13] Willem, “Balancing cube,” Feb 2024. [Online]. Available: <https://willemppennings.nl/balancing-cube/>
- [14] “Ec 45 flat Ø43.5 mm, brushless, 50 w, with hall sensors.” [Online]. Available: https://www.maxongroup.com/maxon/view/product/motor/ecmotor/ecflat/ecflat45/651606?etcc_cu=onsite&etcc_med=Header+Suche&etcc_cmp=mit+Ergebnis&etcc_ctv=Layer&query=ec+45+50w
- [15] N. Corporation, “Cmc_24h series.” [Online]. Available: <https://www.nidec.com/en/product/search/category/B101/M102/S100/NCJ-24H-24-01/>
- [16] Q. Zhang, “Self-balanced cube robot modeling and control,” Columbia University, Department of Electrical Engineering, Course Project Report EE6602 Project, 2023, accessed: [insert access date]. [Online]. Available: https://www.columbia.edu/~ja3451/courses/E6602/projects/balanced_cube.pdf
- [17] M. Muehlebach, M. Gajamohan, and R. D’Andrea, “Nonlinear analysis and control of a reaction wheel-based 3d inverted pendulum,” *52nd IEEE Conference on Decision and Control*, pp. 1283–1288, 2013. [Online]. Available: <https://api.semanticscholar.org/CorpusID:11813217>
- [18] R. Singh, V. K. Tayal, and H. P. Singh, “A review on cubli and non linear control strategy,” *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, pp. 1–5, 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:18372358>

Appendix A

Addenda

Graduate Attribute (GA) Responses

Graduate Attribute	Student Response
GA 1: Problem Solving	
Student Response:	For my self-balancing cube project, I began by analysing the problem: how to maintain the balance of a cube on a point. I began by looking at what others had done and the methods they had used. I then proceeded by analysing the dynamics of the cube, including its moments of inertia and the torques produced by the reaction wheels. This was then modelled in MATLAB and a Simulink Multibody simulation. This simulation allowed me to test various controller parameters and motor specifications. From this, I could reach sensible conclusions about the feasibility of the design before beginning to build a physical prototype. The simulation was critical for understanding how different design choices, such as motor speed and torque, would impact the overall system stability.
GA 4: Investigations, Experiments, and Data Analysis	

Graduate Attribute	Student Response
Student Response:	I've conducted a thorough investigation into the self-balancing problem using a physics simulation as my main research tool. This simulation, which I built in MATLAB/Simulink, acted like a controlled experiment where I could test different ideas. The simulation allowed me to see how the cube would react to different forces, which helped me find the best control strategy, specifically the LQR controller. My analysis of this data showed me what my motors needed in terms of torque and speed, which helped me draw valid conclusions about the physical design of my project.
GA 5: Use of Engineering Tools	
Student Response:	I've used a range of professional engineering tools throughout this project. My work started with MATLAB and Simulink Multibody, which were essential for creating a model of the cube's dynamics and designing my LQR controller. The simulation allowed me to predict how the system would behave before building any hardware. On the hardware side, my use of the ESP32 and the implementation of bidirectional DShot were a key part of the project. I encountered some challenges with DShot communication, and to solve them, I had to use an oscilloscope to analyse the data stream. This allowed me to visualise the signals and identify a timing issue that was causing the communication to fail. Additionally, I modified an existing IMU library to be compatible with the ESP32, which further demonstrates my ability to adapt and debug software for new hardware. Finally, parts were designed in CAD and then 3D printed. This saved time and costs and allowed me to change the design when required.
GA 6: Professional and Technical Communication	

Graduate Attribute	Student Response
Student Response:	The final report is a comprehensive technical document, detailing the entire engineering process, from the initial theoretical modelling to the final test results. This report is structured to present the problem, my methodology, the outcomes of my simulations and experiments, and my conclusions in a clear, logical manner. I will also be prepared to deliver an oral presentation that concisely summarises my findings.
GA 8: Individual Working	
Student Response:	This work has been individual for all aspects of this project. From the initial research and project planning to the hardware design and software development, all work has been conducted independently. This included learning new concepts, troubleshooting problems, and managing my project timeline to ensure key milestones are met.
GA 9: Independent Learning Ability	
Student Response:	This project has been a significant exercise in independent learning. I had to teach myself complex topics like LQR control and the bidirectional DShot protocol, as these were not part of my coursework. When I faced technical problems, like needing to adapt the IMU library for the ESP32, I took the initiative to find solutions on my own.

A.1 Ethics Forms



PRE-SCREENING QUESTIONNAIRE OUTCOME LETTER

STU-EBE-2025-PSQ001918

2025/09/08

Dear Griffin Trace,

Your Ethics pre-screening questionnaire (PSQ) has been evaluated by your departmental ethics representative. Based on the information supplied in your PSQ, it has been determined that you do not need to make a full ethics application for the research project in question.

You may proceed with your research project titled:

Self-Balancing Cube of Awesomeness

Please note that should aspect(s) of your current project change, you should submit a new PSQ in order to determine whether the changed aspects increase the ethical risks of your project. It may be the case that project changes could require a full ethics application and review process.

Regards,

Faculty Research Ethics Committee