

DEVELOPMENT OF A REACTION WHEEL SYSTEM FOR CAL POLY'S
SPACECRAFT ATTITUDE DYNAMICS SIMULATOR

A Thesis
presented to
the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Aerospace Engineering

by
Caleb Nalley
June 2025

© 2025
Caleb Nalley
ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Development of a Reaction Wheel System for Cal Poly's Spacecraft Attitude Dynamics Simulator

AUTHOR: Caleb Nalley

DATE SUBMITTED: June 2025

COMMITTEE CHAIR: Eric Mehil, Ph.D.
Professor of Aerospace Engineering

COMMITTEE MEMBER: Kira Abercromby, Ph.D.
Professor of Aerospace Engineering

COMMITTEE MEMBER: Richard Milford, Ph.D.
Part-Time Lecturer of Aerospace Engineering

COMMITTEE MEMBER: Erin Pearse, Ph. D.
Professor of Mathematics

ABSTRACT

Development of a Reaction Wheel System for Cal Poly's Spacecraft Attitude Dynamics Simulator

Caleb Nalley

The Cal Poly Spacecraft Attitude Dynamics Simulator (SADS) is an ongoing project to develop a system capable of simulating the attitude dynamics and kinematics of a rigid body in space. Actuations on the SADS are performed by reaction wheels, however the current reaction wheels have various issues that decrease system performance and are not independent of the SADS. The reaction wheel developed as part of this thesis will be designed to improve performance and set out design specifications tailored to the SADS, serve as an independent assembly that can be transferred to other systems, and provide a general outline for reaction wheel development for any system. The reaction wheel system developed as part of this thesis is not meant to be a state-of-the-art reaction wheel or implement new technologies, rather develop a reaction wheel system in a low-cost educational environment.

The reaction wheel was designed to provide a momentum capacity of 0.500 N m s , a wheel inertia of $1.698 \times 10^{-3} \text{ kg m}^2$, a maximum spin rate of 2760 rpm, and an SADS angular acceleration of $7.23 \times 10^{-5} \text{ rad/s}^2$. In testing, the measured values resulted in a momentum capacity of 0.463 N m s (7.36 % below the design value), a wheel inertia of $1.716 \times 10^{-3} \text{ kg m}^2$ (1.08 % above the design value), and a top speed of 2577 rpm (6.63 % under the target). The SADS angular acceleration was $9.04 \times 10^{-3} \text{ rad/s}^2$, exceeding the angular acceleration requirements.

ACKNOWLEDGMENTS

Thanks to:

- Dr. Mehiel for his guidance and support throughout this thesis,
- Sage Alling for his help in the manufacturing process,
- Bricen Rigby and Cameron Zorio for their assistance with testing,
- My friends, who encouraged me throughout this thesis,
- My family, who have shown unwavering support and love throughout my education.

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF SYMBOLS	xi
CHAPTER	
1. INTRODUCTION	1
1.1 Literature Review	1
1.1.1 Dynamics Simulators and Reaction Wheel Background	1
1.1.2 Spacecraft Attitude Dynamics Simulator Background	5
1.1.3 Manufacturing Reaction Wheels	9
1.1.4 Modeling Reaction Wheel Performance	13
1.2 Thesis Objectives	18
2. DESIGN REQUIREMENTS	19
2.1 Geometry Constraints	19
2.2 Momentum Requirements	21
2.2.1 Industry Capabilities	22
2.2.2 Motor Selection	23
2.2.3 Inertia of the Wheel	25
2.3 Torque Requirements	31
2.4 Design Requirements Conclusion	38
3. HARDWARE AND SYSTEM DEVELOPMENT	40
3.1 Design	40
3.2 Manufacturing	45

3.3	Electronics	46
4.	SOFTWARE AND TESTING	52
4.1	Software	52
4.2	Validating Design Specifications	55
4.2.1	Testing Inertia of the Wheel	55
4.2.2	Testing Momentum Capacity of the Wheel Mounted on SADS	57
4.2.3	Torque Profile Test	62
4.2.4	Sources of Error	63
4.3	Control Demonstration	65
4.4	Software and Testing Conclusion	67
5.	CONCLUSION	70
	BIBLIOGRAPHY	72
	APPENDICES	
A.	List of Acronyms	76
B.	Code	77

LIST OF TABLES

Table	Page
1.1 Component Weights for MBRW [22]	12
2.1 Specifications of the RWp500 Reaction Wheel [20]	23
2.2 Motor Specifications [2]	24
2.3 Key Reaction Wheel Design Specifications	39
3.1 Bill of Materials	44
4.1 Key Reaction Wheel Tested Specifications	69

LIST OF FIGURES

Figure	Page
1.1 Line of Sight Pointing Errors and Resulting Effects on Image Quality [4]	3
1.2 Pyramidal Reaction Wheel Diagram [16]	4
1.3 Mk. III System Diagram [13]	6
1.4 Mk. I and Mk. II of the SADS [13]	7
1.5 Previous SADS Reaction Wheel	8
1.6 Mk. III of the SADS [13]	9
1.7 SADS Block Diagram	10
1.8 Student Produced Reaction Wheel [3]	11
1.9 Structural dynamic modes of a reaction wheel [16]	15
2.1 SADS Mounting Surface	20
2.2 RWp500 developed by Blue Canyon Technologies [20]	22
2.3 Open/Closed Loop Control [5]	25
2.4 Cross Section of Simplified Shape of Reaction Wheel	26
2.5 h_2 as a Function of r_o , I_T , and h_1	29
2.6 Fixed h_2 Plot from fig. 2.5	30
2.7 Satellite in LEO Tracking a Ground Point	34
2.8 Updated Axis for Pyramidal Reaction Wheel System	36
3.1 Exploded Views of Reaction Wheel Assembly	42
3.2 Assembled Views of Reaction Wheel Assembly	43
3.3 Sliced View of Reaction Wheel Assembly	43
3.4 Required Electronic Components [5]	44
3.5 Manufactured Assembly Components	46

3.6	Manufactured Assembly	47
3.7	Additional Electronic Components	47
3.8	Non-Inverting Operational Amplifier [19]	49
3.9	Single Reaction Wheel Electronics Configuration for Testing	50
3.10	CANopen Possible Configuration for SADS	51
4.1	Voltage to Torque Diagram	54
4.2	NanoJ Operating Frequency [5]	54
4.3	Plots for Determining Inertia	56
4.4	Zoomed In View of Figure 4.3b to Show Effects of Static Friction . . .	57
4.5	SADS Testing Platform	58
4.6	Motor Data for SADS Test	60
4.7	IMU Data from SADS Test	61
4.8	Back EMF Comparison	62
4.9	Plots for Torque Profile	64
4.10	SACS Testing Platform	66
4.11	Plots for PID Test on SACS	68

LIST OF SYMBOLS

Constants

μ Standard Gravitational Parameter of Earth

π Pi

ω_E Angular Velocity Scalar of Earth

r_E Radius of Earth

Symbols

α Phase

ρ Density

τ Torque Scalar

$\boldsymbol{\tau}$ Torque Vector

ω Angular Velocity Scalar

$\boldsymbol{\omega}$ Angular Velocity Vector

$\dot{\omega}$ Angular Acceleration Scalar

$\boldsymbol{\dot{\omega}}$ Angular Acceleration Vector

C Amplitude Coefficient

D Disturbance Force or Torque

\mathbf{e} Body Fixed Reference Frame Vector

f Disturbance Frequency

\mathbf{F} Force Vector

h Height

H Angular Momentum Scalar

\mathbf{H} Angular Momentum Vector

I Inertia Scalar

\mathbf{I} Inertia Tensor

m Mass

n	Number of Harmonics
r	Radius
R	Resistance
s	Harmonic Number
t	Time
V	Volume
V	Voltage
\mathbf{w}	Wheel Spin Axis Vector
\mathcal{W}_4	Reaction Wheel to Body Transformation Matrix
\mathbf{x}	Wheel axis to Center of Mass
\hat{x}	Body Axis Unit Vector
\hat{y}	Body Axis Unit Vector
\hat{z}	Body Axis Unit Vector

APPENDICES

Chapter 1

INTRODUCTION

1.1 Literature Review

1.1.1 Dynamics Simulators and Reaction Wheel Background

As spacecraft usage continues to grow and new technology is developed for space-based systems, there is increased demand to test spacecraft prior to launch. Once a system is in outer space, it is often extremely expensive and difficult to recover it or conduct repairs, making ground testing a crucial component in spacecraft development. This has led to the creation of ground based systems that allow testing of a spacecraft to ensure nominal operation. One such category of spacecraft testing systems are dynamics simulators, which allow for testing of the dynamics of a system that mimic behavior in the space environment. The topic of this thesis is one such system, the Cal Poly Spacecraft Attitude Dynamics Simulator (SADS), which aims to simulate the attitude dynamics and kinematics of a rigid body in space [18].

Being able to accurately control the attitude of a spacecraft is crucial in many missions, and has lead to the development of a myriad of control systems for spacecraft. The focus of this thesis will be in relation to reaction wheels as a control mechanism. Reaction wheels operate by storing rotational energy and imparting an internal torque on a system, as opposed to other control mechanisms that utilize external torque to control a system, such as thrusters or propellants. Reaction wheels have long been used as a control method in the aerospace industry, and continues to be a reliable technology for controlling modern spacecraft. Reaction wheels fall into a category of

momentum-exchange devices (MED), that includes other technologies such as control-moment gyros (CMG). The focus of this literature review and thesis will be on topics related to developing reaction wheels and verifying their capabilities with a space dynamics simulator. Consequently, alternative MEDs will not be considered in this thesis.

When three or more reaction wheels are introduced to a system, it becomes possible to have 3-axis control of a spacecraft. The SADS utilizes four reaction wheels in a pyramidal configuration, which will be discussed later.

As imaging technology capabilities increase, there is a need for greater accuracy guidance, navigation, and control systems, and thus verifying reaction wheel pointing capability is an important system checkpoint for any imaging mission. According to a survey on reaction wheel disturbances for line-of-sight jitter analysis, missions such as the Habitable Exoplanet Observatory which aims to directly image Earth-sized exoplanetary systems around Sun-like stars must be designed to manage line of sight jitter to within 0.1 to 0.5 milli-arcseconds (4.85×10^{-10} to 2.42×10^{-9} rad) to achieve the desired exoplanet image quality [8]. Jitter refers to unwanted small, high-frequency disturbances or vibrations introduced into a spacecraft's attitude or pointing due to imperfections or dynamic imbalances in the reaction wheel system. The importance of line of sight pointing errors and how it affects the resulting image is shown in fig. 1.1. This is one of the functions of the SADS, as it would allow for verification of pointing accuracy when reaction wheels capable of precise movement are utilized and properly calibrated. The exact capabilities are still in development the various subsystems continue to be improved by students at Cal Poly.

In addition to the reaction wheels, it is necessary to have a control law that dictates how the fast the wheels are accelerated/de-accelerated to control the spacecraft. There are various control laws that exist, and selecting one depends on the objec-

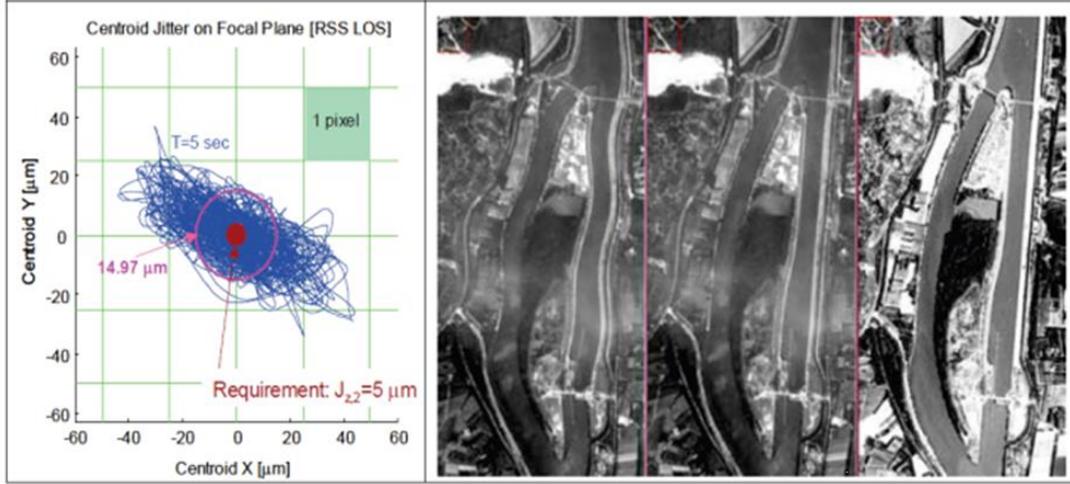


Figure 1.1: Line of Sight Pointing Errors and Resulting Effects on Image Quality [4]

tives of system performance. Different control laws produce different wheel speeds, accelerations, system settling time, and more. Another one of the goals of the SADS is to be able to test and verify various control laws by analyzing the behavior of the SADS as the control laws are implemented. Control laws will be discussed further in the actual thesis, however testing and developing control laws is not the objective of this work.

As mentioned above, the SADS is designed to incorporate four reaction wheels in a pyramidal configuration, as shown in fig. 1.2. Three reaction wheels allows for full 3-axis attitude control, but adding additional wheels adds redundancy. Although reaction wheels are a fairly well-proven technology, there are still cases of reaction wheel failure on spacecraft, such as when several reaction wheels failed on the Kepler system in 2013 [6]. This emphasizes the importance of having robust reaction wheels, as well as having redundancy in the reaction wheels. In the pyramidal configuration, one wheel can fail and the spacecraft will still be able to maintain 3-axis attitude control, however if two or more fail, the spacecraft will lose full control capability.

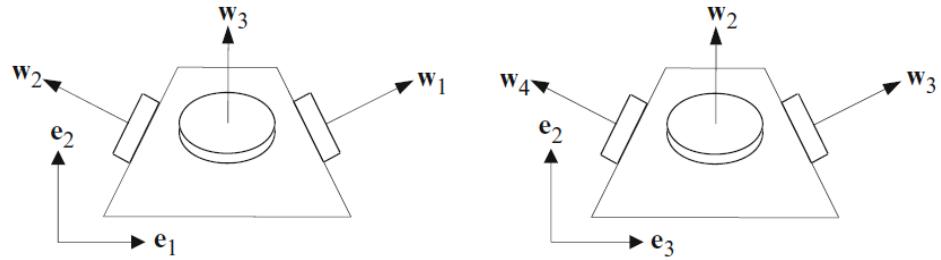


Figure 1.2: Pyramidal Reaction Wheel Diagram [16]

Research has been conducted on operation procedures for spacecraft in the case of multiple reaction wheel failures, but it will not be covered in this literature review.

Reaction wheels come with various different capabilities, such as maximum torque and maximum rotational speeds. Common metrics used to determine reaction wheel capabilities and quality are peak torque (N m) and momentum capacity (N m s). A large variety of capabilities are commercially available, however many of them fall outside of the range required and reasonable for this project. The SADS would fall into the category of requirements similar to a small-sat (satellites weighing in the 100 kg or less range). According to a collection of state-of-the-art small-sat Guidance, Navigation, and Control (GNC) components compiled by NASA, a typical small-sat reaction wheel can have peak torques typically ranging from ~ 0.00023 to ~ 0.3 Nm, and momentum capacities ranging from ~ 0.0005 to ~ 8 Nms [1]. For a more general range not specific to small-sat applications, peak torques typically range from ~ 0.01 to ~ 1.0 Nm, and momentum capacities range from ~ 2 to ~ 250 Nms [16]. One of the primary goals of this thesis is to make quality reaction wheels capable of actuating the SADS at desirable rates for a system that simulates a small satellite, which requires preliminary computations to ensure that it has the specifications such as momentum capacity and torque to execute the desired rate. These calculations will be performed in chapter 2, and the numbers obtained in that section will fall into the specification ranges discussed here.

1.1.2 Spacecraft Attitude Dynamics Simulator Background

The work of this thesis will contribute to the Cal Poly Spacecraft Attitude Dynamics Simulator, an ongoing project overseen by Dr. Eric Mehiel of the Cal Poly Aerospace Department. The aim of this system is to simulate the attitude dynamics and kinematics of a rigid body in space [18]. The system first performed attitude control in 2010 [14], and has been continuously updated. The system has been the focus of many past theses, and the work presented in this thesis will further contribute to the systems development and improvement by developing high-quality reaction wheels.

The work is also continued through a club at Cal Poly SLO called PolySim, headed by Dr. Mehiel. PolySim aims to integrate hardware and control theory to develop systems capable of recreating many of the technologies and techniques utilized on spacecraft. These systems also allow experimentation and verification of concepts through testing. This work is also used as educational material for students learning control theory, providing a hands-on example of content learned in class. The reaction wheel developed as part of this thesis aims to further demonstrate the control capabilities utilized in modern spacecraft, as well as act as an educational tool.

The SADS is made up of multiple systems that work together to provide a fully functional system, as shown in fig. 1.3. The focus of this section of the literature review will be with respect to the pyramidal reaction wheel platform (PRWP). The PRWP is comprised of four reaction wheels arranged in a pyramidal platform as seen in fig. 1.2, and is mounted on a spherical air bearing [13]. As shown in fig. 1.3, the reaction wheels are included as a part of the actuator system, and interact with the command and data handling system and the power system.

The first iteration of the PRWP was named Mk. I, shown in fig. 1.4, and was developed by Middlestead [18]. Mk. I had individual components that were functional but was

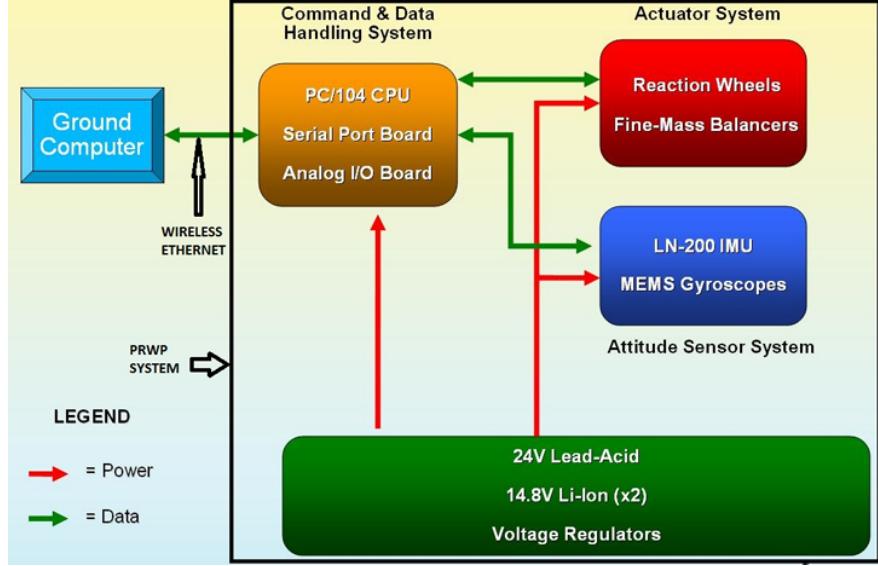
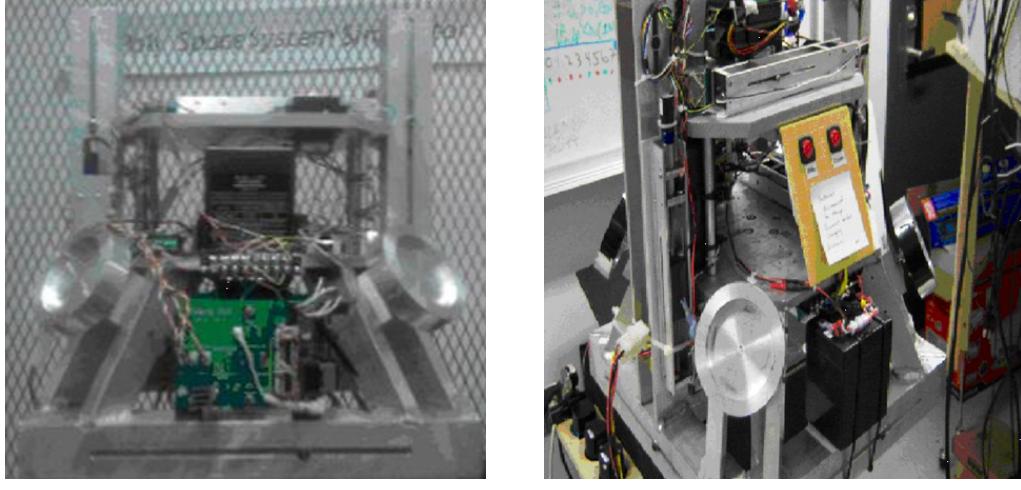


Figure 1.3: Mk. III System Diagram [13]

incapable of executing real-time tests. The limits of the data processing capability of the Bluetooth link and onboard CPU caused these issues, but were addressed by the work of Kinnett [14] and Downs [9], which resulted in the first functional closed-loop attitude control in real time. This second iteration was named Mk. II, which is also shown in fig. 1.4. Mk. II was marred by electrical unreliability which led to incidents of electrical failure. Mk. III, pictured in fig. 1.6, aimed to address these issues, as well as adding new components and subsystems to improve the capabilities of the overall subsystem. Recent work included on Mk. III was done by Kato, which adopted the MATLAB xPC kernel for better real-time hardware control, and integrated the Litton LN-200 Inertia Measurement Unit (IMU) as the new attitude sensor [13]. Work done by Dam used methods to reduce undesired external torque from the system via System Identification and identify mass properties of the system [7]. A new subsystem was included with the addition of an automatic mass balancing system to reduce alignment errors between the center of mass and the center of rotation, as well as provide an inertia tensor for use in software [11]. If the center of rotation is aligned with the center of gravity of the pyramid structure, no resultant gravity torque acts upon the

system, which reduces the effort required by the reaction wheels to actuate the system. The mass balancing system is an ongoing project, with current students working to improve its capabilities.



(a) *Mk. I*

(b) *Mk. II*

Figure 1.4: *Mk. I and Mk. II of the SADS [13]*

Mk. III has existing reaction wheels from prior works, however the quality and implementation of these wheels created several issues. Due to imperfections in the reaction wheels, a drift torque was introduced into the system and caused increased controller efforts and changes in the plant model. This lead to the reaction wheels reaching saturation speed more rapidly, allowing for less operating time overall. The reaction wheels also noticeably vibrated while operating, which was likely caused by improper mating of the motor shaft and the wheel, as well as poor balancing. The machining of these wheels were not precise either, such as the stock finish being left as the outer diameter of the wheel instead of machining it down to a precise radius. Addressing these issues would help decrease the drift torque [13], and will be a critical goal in the development of reaction wheels done in this thesis. The specific effects that stem from imperfections in the reaction wheels and lead to decreased performance will be further discussed in section 1.1.4.

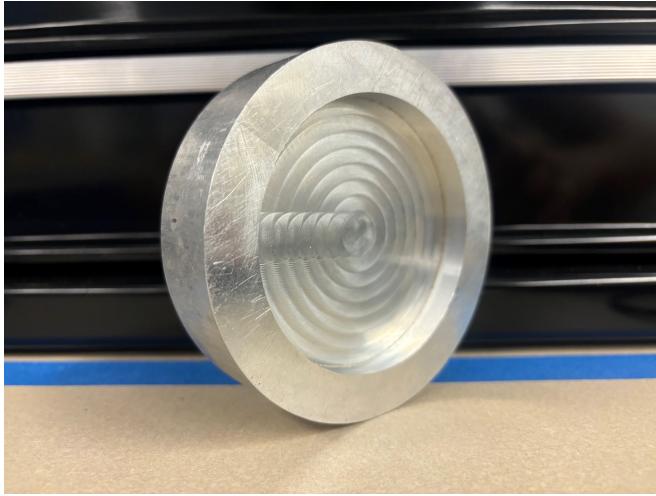


Figure 1.5: Previous SADS Reaction Wheel

Improving the reaction wheel design leads to greater control capabilities, smaller settling times, and longer operational times before saturation speed is met. Additionally, this more in-depth design could improve mass optimization of the reaction wheel, decrease vibration, and increase system compactness, as the current reaction wheel design is fairly simple as seen in fig. 1.5.

Additionally, these old reaction wheels were designed specifically to be mounted on the SADS, requiring the pyramidal structure to hold the motor and placing incorporated electronics on the SADS system itself. This resulted in the old reaction wheels relying on the SADS structure for assembly and functionality. An important design change as a part of this thesis is to develop an independent reaction wheel system, one that is self-contained and could be moved from system to system. This reaction wheel system will contain the reaction wheel, the housing needed to support and contain it, and the associated electronics. It will have mounting holes available for attachment to the SADS or other subsystems.

The current system architecture has been created by current students working on the SADS and is shown in fig. 1.7. The reaction wheels fall under the attitude control system, and will have to communicate with the flight computer. Data for actuations will

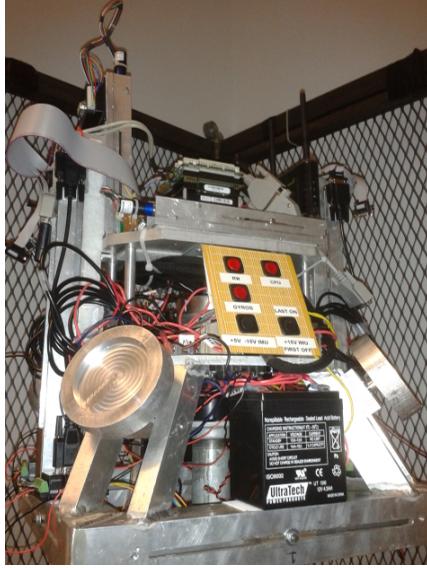


Figure 1.6: Mk. III of the SADS [13]

be generated from the attitude determination system, and there are several different modes that can be created to demonstrate various different functionalities.

As each subsystem is still underdevelopment, future work will bring together the works of current and past students to have a fully functional SADS as conceptualized in fig. 1.7. The reaction wheel developed as a part of this thesis will serve as a foundation for the attitude control system, and future work will be required to fully implement with the SADS and various other subsystems.

1.1.3 Manufacturing Reaction Wheels

A typical reaction wheel assembly is comprised of a rotating flywheel that is typically supported by ball bearings, as well as an internal brushless DC motor and other electronics [16]. Design choices can be made to develop a wheel for high torque capability versus a wheel with high angular momentum capacity, but will reuse components such as flywheels and bearing assemblies. Manufacturing with high-quality bearings, using lubrication systems, and brushless motors reduce jitter in a reaction wheel and help

Spacecraft Attitude Dynamics Simulator (SADS)

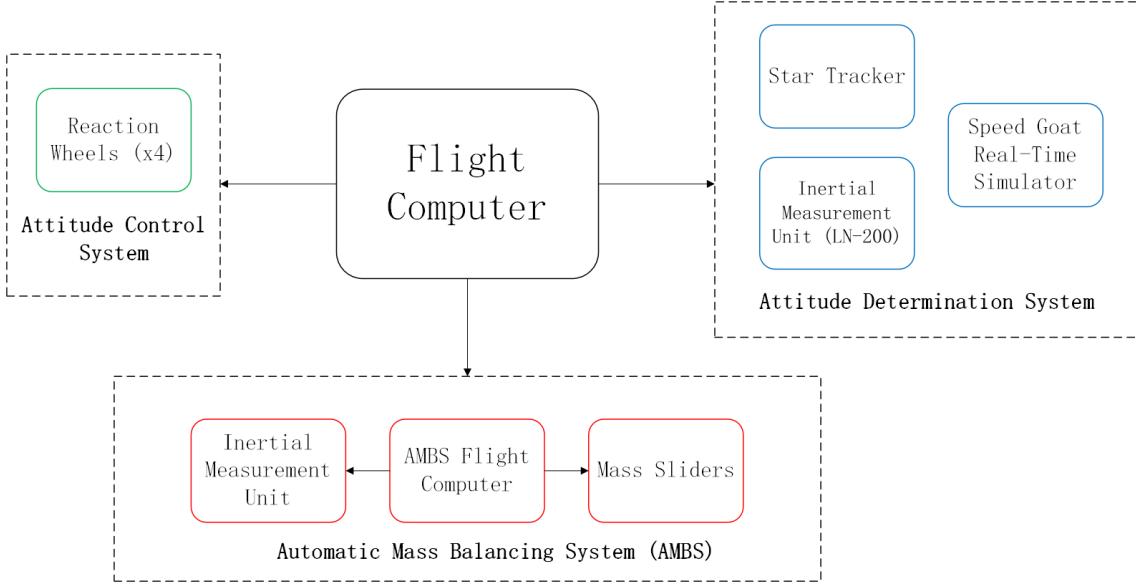
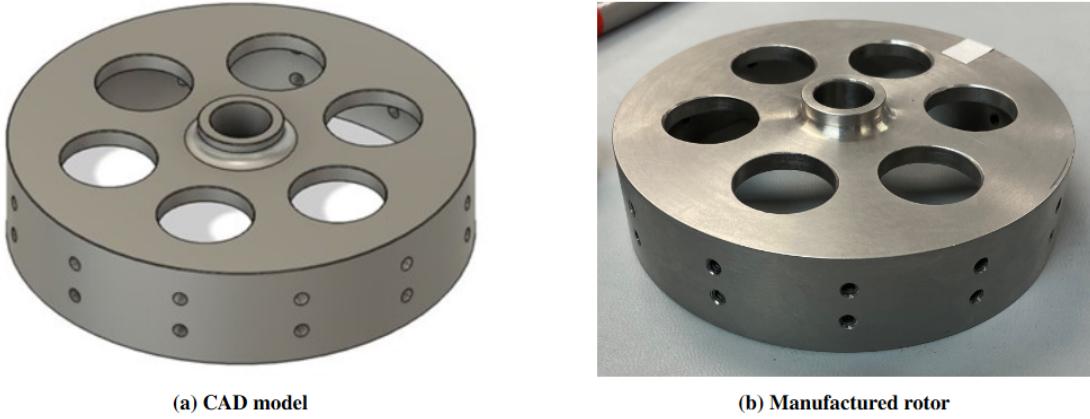


Figure 1.7: SADS Block Diagram

to ensure desired performance in the system. Industry companies utilize these techniques to increase wheel performance [1]. Jitter will be discussed in further detail in section 1.1.4.

In order to ensure the reaction wheel is operating as desired, electronics are incorporated into the design to measure wheel speed. A digital or analog tachometer can be used to provide a measuring of the wheel speed, and can be implemented with an internal closed-loop controller to hold the commanded wheel speed [16]. Some motors also offer built-in components that can perform this functionality, such as hall sensors or encoders. This can simplify the required electronics by having both the motor and required electronics for measuring combined together in one system, making it an advantageous design choice when selecting motors. This will be further discussed in section 2.2.2



(a) CAD model

(b) Manufactured rotor

Figure 1.8: Student Produced Reaction Wheel [3]

Finding the specifics of current industry standard reaction wheel manufacturing information is challenging, due to the proprietary nature of the information. There has been similar work to develop student made reaction wheels, however many of these applications have been for satellite applications rather than simulator applications. Burkhart, Snider, and Firth developed low cost student manufactured and balanced reaction wheels for a small satellite [3]. They utilized 304 stainless steel for the wheel, and followed a standard design choice of concentrating mass along the wheel's rim to maximize moment of inertia relative to mass, pictured in fig. 1.8. The also included two sets of 12 holes along the wheel's rim to allow for threaded screws to be inserted into the wheel. This was done for the purpose of balancing the wheel, as slight adjustments in the screw lengths could effectively balance the wheel and correct for any errors in the manufacturing process or improper balancing as mentioned in section 1.1.4. The group was also able to calculate the imbalance location using MATLAB and Arduino to be able to pinpoint the necessary place to adjust wheel balancing, which is explained in detail in [3].

One publicly available report on developing reaction wheel was released by NASA JPL. This report focused on the development efforts for magnetic bearing reaction wheels (MBRW), as compared to the typical ball bearing. NASA JPL found that this

design was applicable for interplanetary and orbital spacecraft without “appreciable penalties on weight and power” [22]. The magnetic bearing provided many benefits relative to the ball bearing, including low drag torque operation over the entire speed range (including through zero speed) and increased redundancy (no single point failures). Component parts and their associated weights were tabulated in the JPL MBRW report, which includes parts specific to a magnetic bearing and also general parts required for reaction wheels. The table has been included in table 1.1 to provide a typical component list for a reaction wheel assembly. The total assembly weight of the MBRW was 8.9 lbs.

Table 1.1: Component Weights for MBRW [22]

Component	Component Weight (lbs)
Rotor Assembly	3.2
Shaft	0.09
Rim-Web-Hub	1.74
Magnetic Bearing Rotors	0.11
Motor Cage	1.19
Miscellaneous	0.07
Magnetic Bearing Stator Assembly	1.34
Iron Parts	0.40
Coil	0.11
Magnet	0.12
Spacer	0.04
Housing Assembly	4.0
Lower Housing	1.26
Upper Housing	0.75
Motor Stator	1.36
Cover	0.21
Tachometer Probe Supports	0.13
Touchdown Bearing Support	0.11
Tachometer Probes and Miscellaneous	0.18

The development of reaction wheels for this thesis will not take the form of MBRWs, but the information provided by NASA JPL is valuable and there are many correlations between MBRWs and traditional ball bearing systems.

These examples of reaction wheel systems serve as a baseline of the components and methods involved in developing these systems, and serve as a point of reference for the system developed as a part of this thesis.

1.1.4 Modeling Reaction Wheel Performance

When designing a reaction wheel, it is important to understand the dynamics involved in modeling a reaction wheel. This provides information that influences designing the wheel in a way that avoids or minimizes possible sources of error. One of the most important considerations in modeling reaction wheels is accounting for the disturbances produced by reaction wheels. These disturbances, often referred to as jitter, can be transferred to the spacecraft, and result in degraded control accuracy. There are many possible sources for disturbances on a spacecraft, but jitter due to the reaction wheels are anticipated to be the largest contributor [17]. Understanding and modeling this jitter is of great importance for developing reaction wheels, and several methods have been developed to achieve this.

Reaction wheels and their associated equations of motion often assume that they are perfectly balanced, an assumption which simplifies analysis but does not reflect the reality of almost every reaction wheel system. Either due to imperfect machining, improper mounting or other sources of error, imbalance is guaranteed to be present in a reaction wheel system. There are two classes of imbalances discussed, static imbalance and dynamic imbalance [16], each of which will be discussed below.

Static imbalance refers to the reaction wheel's center of mass being located off of the axis of rotation. In order to maintain rotation around the axis of rotation, an external force provided by the spacecraft, through the bearing assembly, acts upon the wheel,

and thus creates an additional force on the spacecraft that was not accounted for in the perfectly balanced case.

The equation for the static imbalance force is:

$$\mathbf{F}_s = m^w (\omega^w)^2 \mathbf{w} \times (\mathbf{w} \times \mathbf{x}) = -m^w (\omega^w)^2 \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} \quad (1.1)$$

where m^w is the mass of the rotor, ω^w is the magnitude of the angular velocity of the rotor, \mathbf{w} is the unit vector along the wheel's axis of rotation, \mathbf{x} is the vector from the rotor's axis of rotation to its center of mass, and $x_1, x_2, 0$ are the components of \mathbf{x} in the coordinate system, with the third component being zero because \mathbf{x} lies in the plane perpendicular to \mathbf{w} .

The last form of eq. (1.1) assumes that $\mathbf{w} = [0 \ 0 \ 1]^T$. The static imbalance force is a radial force that is constant in the wheel frame but rotating with angular velocity ω^w in the spacecraft frame.

Dynamic imbalance occurs when the axis of rotation of the wheel is not along its principal axis. In order to keep the wheel rotating at a constant rate ω^w about the spin axis, which is assumed to be $\mathbf{w} = [0 \ 0 \ 1]^T$, a torque is required to be exerted by the spacecraft on the wheel.

The equation for the dynamic imbalance torque is:

$$\boldsymbol{\tau}_d = \boldsymbol{\omega} \times \mathbf{H} = (\omega^w)^2 \mathbf{w} \times (\mathbf{I}\mathbf{w}) = (\omega^w)^2 \begin{bmatrix} -I_{23} \\ I_{13} \\ 0 \end{bmatrix} \quad (1.2)$$

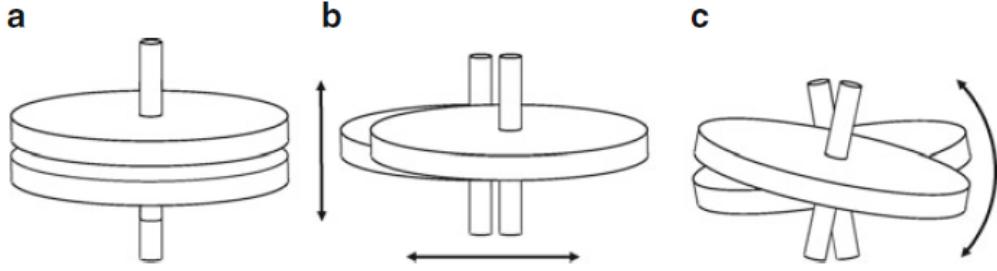


Figure 1.9: Structural dynamic modes of a reaction wheel [16]

where τ_d is the dynamic imbalance torque, ω is the angular velocity vector of the rotor, \mathbf{H} is the angular momentum vector, and \mathbf{I} is the inertia matrix.

This represents a radial moment that is constant in the wheel frame and rotating with angular velocity ω^w in the spacecraft frame.

It is also important to note that in addition to static and dynamic imbalances, there are resulting structural dynamic modes that can result from interactions between the reaction wheel and bearings. These are illustrated in fig. 1.9 and are classified as (a) axial translations, (b) radial translations, and (c) rocking.

Wheel imbalance is typically the largest disturbance present in a reaction wheel system, however there are additional disturbance sources to consider as well [17]. Disturbances can be caused by the ball bearings originating from irregularities in the balls, races, and cage, which can be amplified at the sub- and super-harmonics of the wheel's spin rate. Lubricant dynamics can also come into play and induce low-frequency disturbances and torque. The motor also brings disturbances, such as cogging in the brushless DC motor generating very high-frequency disturbances.

Modeling all of the dynamics associated with these sources of disturbance can be quite complex and computationally heavy, and a better approach is to use a stochastic model to approximate disturbances at discrete wheel speeds. Research done by Hasha

produced an empirical model by running induced vibration tests, and they found that reaction wheel disturbances are tonal in nature i.e. the disturbance frequencies are a linear function of wheel speed [12]. Using the physics of a rotating wheel, and the data associated with the tests, it was possible to model the disturbances as a series of discrete harmonics at frequencies that vary linearly with wheel speed and with amplitudes proportional to the wheel speed squared [17]. This relationship is listed as follows:

$$D(t) = \sum_{i=1}^n C_i \omega^2 \sin(2\pi s_i \omega t + \alpha_i) \quad (1.3)$$

where $D(t)$ is the disturbance force or torque, n is the number of harmonics included in the model, C_i is the amplitude coefficient of the i -th harmonic, ω is the wheel speed, s_i is the i -th harmonic number, and α_i is a random phase assumed uniformly distributed over $[0, 2\pi]$.

The harmonic numbers are non-dimensional frequency ratios that describe the relationship between the i -th disturbance frequency, f_i , and the spin rate of the wheel, ω as follows:

$$s_i = \frac{f_i}{\omega} \quad (1.4)$$

The model parameters h_i , C_i , and n are wheel dependent, and thus would have to be determined as a part of the experiment testing phase with manufactured reaction wheels. This can be achieved with the use of a MATLAB toolbox that analyzes steady state reaction wheel disturbance data to find the amplitude coefficients and harmonic numbers used in eq. (1.3) for the given reaction wheels [17]. This process is detailed by Masterson, Miller, and Grogan for a ITHACO Space Systems E-type reaction wheel,

and they found that this method yielded good estimates of disturbances over most wheel speeds, however had poor correlation for certain speed ranges. They determined this poor correlation resulted from the fact that the empirical model did not account for disturbance amplifications caused by interactions between the harmonics and the structural modes of the wheel [17].

Modeling friction is also important for reaction wheel systems, however can be largely dependent on the specific system in use. Wheel friction (drag) can be modeled as a sum of viscous and Coulomb components as follows [16]:

$$\tau_{\text{drag}}^w = -\tau_v \omega^w - \tau_c \text{sign}(\omega^w) \quad (1.5)$$

τ_v and τ_c are coefficients that can be determined empirically, often exhibiting a temperature dependence [16]. Although this model suffices for many scenarios, it can be inaccurate when dealing with low speeds and crossing through zero speed. A dynamic friction model that incorporates hysteresis is essential for accurately capturing these phenomena. Literature offers a comprehensive review of the challenges in friction characterization, modeling, and the historical evolution of friction models [16]. The Dahl model, introduced in 1968, was the first dynamic friction model to gain widespread application. Since then, numerous models have been proposed, including the LuGre model, developed by research groups in Lund and Grenoble. This model, an enhancement of the Dahl model, accounts for larger frictional forces at low speeds, a phenomenon known as the Stribeck effect or stiction. Monitoring drag torque on reaction wheels in orbiting spacecraft is standard practice, as increasing drag torque often signals potential wheel failure.

1.2 Thesis Objectives

The purpose of this literature review was to provide a comprehensive background on the topics addressed by this thesis, as well as providing insight into methods and relevant information for the creation of a reaction wheel system for the SADS. Additionally, it provides context for the purpose of SADS, an ongoing project to develop a system capable of simulating the attitude dynamics and kinematics of a rigid body in space.

The primary objectives of this thesis are to create a fully operational independent reaction wheel system designed for the SADS, validate the wheel specifications, and create a repeatable methodology that can be utilized for generally creating reaction wheels. A single reaction wheel system will be developed from the work of this thesis, however future work for the SADS will require four reaction wheels, and this thesis will provide background, methodology, and results relevant to future students working on the reaction wheels and the system as a whole. The architecture of the system will also be designed in such a way that will facilitate multiple reaction wheels functioning simultaneously in the future, as will be discussed later in the thesis.

Chapter 2

DESIGN REQUIREMENTS

Sizing the reaction wheel is a critical part of the design process, and influenced many of the design decisions of the developed reaction wheel. While this reaction wheel was specifically made for the SADS, it is important to understand that each system will hold unique design considerations. Thus the methodology utilized in the development of this reaction wheel will be generalized for a wide range of applications, and can be repeated with a different system. The specifics relating to this system will also be provided as justification for design choices as well as an example to follow.

Firstly, the geometry constraints of the system will be analyzed in section 2.1, as that affects many of the overall dimensions of the wheel. Section 2.2 will cover the methodology of developing the required specifications for the system and ensuring that they are satisfied. Finally, section 2.3 will demonstrate that the properly sized reaction wheels will be able to meet torque requirements levied by a hypothetical small satellite usage.

2.1 Geometry Constraints

Many systems will have limited mounting area or volume, as well as overall volume constraints. This places a primary bound on the size of the wheel, as it must be able to be securely attached to the system. On the SADS, there was an approximately 5.5 in mounting length along the diagonal supports, which limited the size of the wheel and the associated housing/mounting mechanisms as seen in fig. 2.1. In order to concentrate mass closer to the center of mass of the SADS (for inertia purposes,

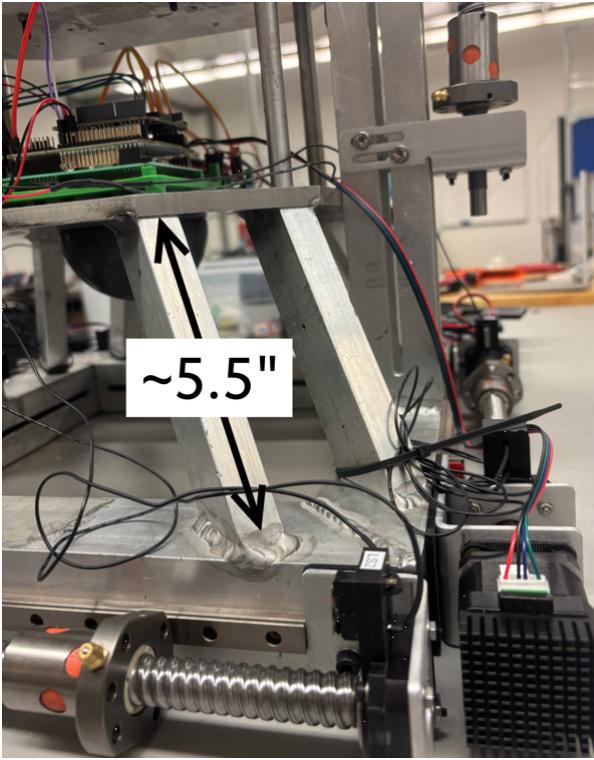


Figure 2.1: SADS Mounting Surface

which will be discussed in section 2.2.3), a primary development goal was to limit the total height of the reaction wheel assembly. For other systems, limiting total assembly height may not be of equal importance, and increasing reaction wheel assembly height can lead to similar reaction wheel capabilities.

Additionally, one of the driving goals of this reaction wheel was to develop a design that would make the reaction wheels, housing, and incorporated circuitry an independent system that could be transferred from system to system. This required having no dependence on the SADS structure, and containing all the necessary reaction wheel components, housing, and connection terminals all in one assembly. An independent system would allow for the created reaction wheels to be transferred from one system to another, as many future systems created by the labs at Cal Poly SLO may require reaction wheels. Additionally, following the steps listed in these sections would allow

for a similar wheel to be reproduced with different mission requirements, without having to conceptualize a new wheel altogether.

2.2 Momentum Requirements

Industry standard reaction wheels are often classified by their angular momentum capabilities, as when reaction wheels are used onboard a system, an important metric is the “saturation” of the wheel. Reaction wheel saturation occurs when a reaction wheel builds up enough stored momentum that its maximum angular velocity, prevents it from providing additional control torque. This requires desaturation via external torques such as magnetorquers or thrusters, which allows the wheel to lower the stored momentum and become operational again. Thus, it is necessary to have enough angular momentum capability to either avoid saturation or have a sufficient system capability that can also be desaturated in a given time frame via external torques. Angular momentum is defined by:

$$\mathbf{H} = \mathbf{I}\boldsymbol{\omega} \quad (2.1)$$

for a 3-axis system, where \mathbf{H} and $\boldsymbol{\omega}$ are vectors representing momentum and angular velocity respectively, and \mathbf{I} represents the inertia tensor of the system. For system design and specification purposes of a single reaction wheel, it is beneficial to perform analysis by reducing eq. (2.1) to a single axis, resulting in eq. (2.2), with scalar values for angular momentum H_{wheel} , wheel inertia I_{wheel} , and wheel speed ω_{wheel} .

$$H_{wheel} = I_{wheel} \omega_{wheel} \quad (2.2)$$

Since inertia of the wheel I_{wheel} will be fixed and there is an upper limit on the wheel speed ω_{wheel} , it is possible to quantify the angular momentum capabilities of the developed system if those are known. Section 2.2.1 will give context to industry standard momentum capabilities that can be applied to our system design.

2.2.1 Industry Capabilities

As mentioned in the literature review, there exists many commercially available reaction wheels produced in industry. Many different sizes, designs, and capabilities of wheels exist, so it is necessary to narrow down the range of industry reaction wheels which will be compared to the system under development. The SADS falls into the category of “small satellites” in terms of sizing and mass, thus it is pertinent to look at reaction wheels developed for this class of spacecraft. Many reaction wheels fall into this category, one of which is the RWp500 Reaction Wheel developed by Blue Canyon Technology pictured in fig. 2.2 [20]. Throughout the design process, this wheel was often referenced for the capabilities and specifications that could be achieved by industry. The specifications are listed in table 2.1.

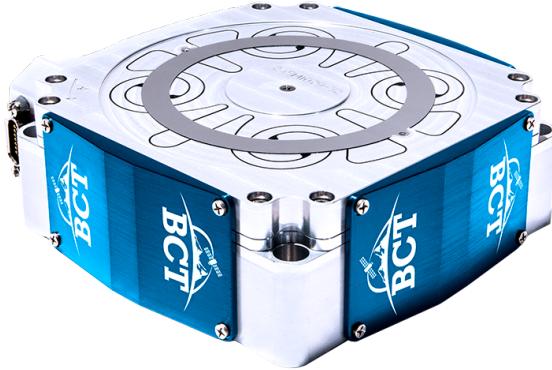


Figure 2.2: RWp500 developed by Blue Canyon Technologies [20]

The most important specification that affected the design of our system was the maximum momentum capability of 0.500 N m s. This value was set as the desired

Table 2.1: Specifications of the RWp500 Reaction Wheel [20]

RWp500 Specifications	
Max Momentum	0.500 N m s
Max Torque	0.025 Nm
Static Unbalance	< 6 g·mm
Dynamic Unbalance	< 180 g·mm ²
Mass	0.86 kg
Dimensions	112 × 124 × 38 mm
Supply Voltage	22–34 VDC
Power at Max Momentum	< 6 W
Peak Power	25 W

momentum capability of the reaction wheel system. Thus, H_{wheel} has been determined for the system, which leaves the designer to choose a combination of inertia and wheel speed that satisfies the momentum requirements H_{wheel} . The wheel speed ω_{wheel} will be determined by the purchased motor, as the known wheel speed capabilities of the motor will translate to wheel speed as discussed in section 2.2.2. With ω_{wheel} determined, it is possible to determine I_{wheel} for the reaction wheel, derived in section 2.2.3.

2.2.2 Motor Selection

One of the driving design considerations was to make the reaction wheel system as compact as possible. For this reason, the category of motors examined for this reaction wheel were compact “flat” motors. In addition, the advantages that brushless DC motors offer such as increased speed control and acceleration capabilities, made them the desirable choice over brushed DC motors. This narrowed down the motor search to flat brushless DC motors.

In order to satisfy the desired 0.500 N m s momentum capability, the motor had to be capable of producing satisfactory wheel speed ω_{wheel} . Without having an exact value for the inertia of the wheel I_{wheel} , it is not possible to determine an exact

ω_{wheel} value to meet the 0.500 N m s momentum capability. Thus, torque abilities were the next consideration in selecting the motor. Torque was commonly specified in industry standard reaction wheel specifications, which allowed torque requirements to be set. The specific capabilities required will be discussed in section 2.3, as well as how to validate the torque capabilities. However, initial torque requirements were driven by industry reaction wheel capabilities. For reaction wheels with 0.500 N m s momentum capabilities, the typical torque capabilities were around 25 mN m . Thus motors were narrowed down to those meeting this torque capability.

Several motors were considered in the initial phases of the thesis, and ultimately the Nanotec DF32M024027-A - Brushless DC flat motor [2] was chosen for torque capabilities, speed, and cost effectiveness. The specifications of the motor are listed in table 2.2, but most importantly the motor has a rated torque of 25.4 mN m , satisfying the torque requirements.

Table 2.2: Motor Specifications [2]

Parameter	Value
Rated Current	0.5 A
Rotor Inertia	35 g cm ²
Peak Torque	76.3 mN m
Rated Torque	25.4 mN m
Peak Current	1.5 A
Rated Voltage	24 V
Rated Power	7.4 W
Rated Speed	2760 rpm

With the motor chosen, this fixed the maximum possible wheel speed ω_{wheel} . As seen in table 2.2, the rated speed is 2760rpm, which directly translates to ω_{wheel} . This left the inertia of the wheel I_{wheel} to be designed to meet the desired angular momentum of 0.500 N m s. Using eq. (2.2), $H_{wheel} = 0.500 \text{ N m s}$, and $\omega_{wheel} = 2760 \cdot \frac{2\pi}{60} \text{ rad/s}$, $I_{wheel} = 1.73 \times 10^{-3} \text{ kg m}^2$.

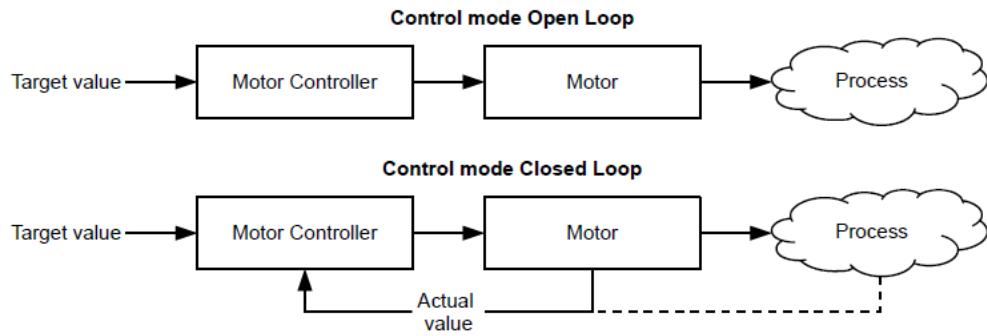


Figure 2.3: Open/Closed Loop Control [5]

The motor also had three hall sensors included as a part of the incorporated circuitry, which was advantageous for control purposes. These hall sensors provided data on motor position, velocity, and torque. It also allowed the controller to utilize closed-loop control, as pictured in fig. 2.3, simplifying the control requirements on the system designer and leading to increased accuracy in motor performance. These hall sensors were also later utilized for data collection of motor performance.

Additionally, the chosen motor required additional electronics such as a motor controller and adapter boards, however that will be discussed in more detail in chapter 3. These were obtained from Nanotec as well, as they were designed to function with the selected motor.

2.2.3 Inertia of the Wheel

The general geometry chosen for the reaction wheel is a cylinder with hollowed out portions as seen in fig. 2.4.

This design was chosen to maximize inertia, minimize mass, while providing structural strength and platforms for assembling the various parts of the reaction wheel system together. It is known that concentrating more of the mass away from the axis of

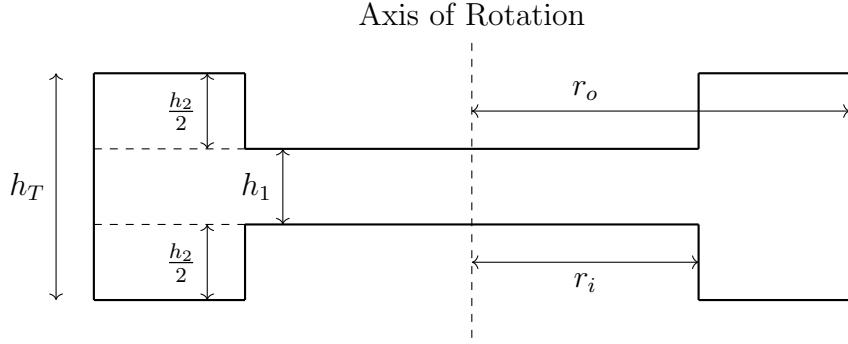


Figure 2.4: Cross Section of Simplified Shape of Reaction Wheel

rotation I_{hoop} provides a larger inertia than a constant disk I_{disk} , as demonstrated from the equations of inertia for each respectively eq. (2.3), eq. (2.4):

$$I_{disk} = \frac{1}{2}mr^2 \quad (2.3)$$

$$I_{hoop} = mr^2 \quad (2.4)$$

For an annulus, the inertia can be written as follows:

$$I_{annulus} = \frac{1}{2}m(r_o^2 + r_i^2) \quad (2.5)$$

Using the above equations and fig. 2.4, it is possible to derive an equation for the shape of the reaction wheel. The symbols required to derive the inertia of this shape are as follows: I_T represents the total inertia; I_1 and I_2 denote the inertia of the solid disk and the annulus (excluding the solid disk), respectively; h_T is the total height; h_1 and h_2 are the heights of the solid disk and annulus without the solid disk; r_i and

r_o correspond to the inner and outer radii; and m_1 and m_2 indicate the masses of the solid disk and the annulus, respectively.

The total inertia of the wheel pictured in fig. 2.4 can be expressed as the sum of a disk and an annulus, I_1 and I_2 respectively:

$$I_T = I_1 + I_2 = \frac{1}{2}m_1r_o^2 + \frac{1}{2}m_2(r_o^2 + r_i^2) \quad (2.6)$$

Assuming a constant density ρ , mass can be expressed as,

$$m = \rho V \quad (2.7)$$

where V is volume. The goal in this formulation is to reduce the total inertia equation I_T to dependence on geometric dimensions, and this can be accomplished by using this substitution for m .

$$V_1 = \pi r_0^2 h_1 \quad (2.8)$$

$$V_2 = \pi(r_0^2 - r_1^2)h_2 \quad (2.9)$$

Multiplying both V_1 and V_2 by ρ yields m_1 and m_2 . Substituting into eq. (2.6).

$$I_T = \frac{1}{2}(\rho\pi r_0^2 h_1)r_o^2 + \frac{1}{2}(\rho\pi(r_0^2 - r_1^2)h_2)(r_o^2 + r_i^2) \quad (2.10)$$

Because eq. (2.10) depends on four variables (r_o, r_i, h_1, h_2) , it is advantageous to fix a variable to be able to visually represent the resulting three-dimensional surface that shows the combination of variables to achieve the desired inertia. As mentioned in section 2.1, there is a limited mounting space of approximately 5.5 in, limiting the total space for the housing of the wheel and the wheel itself. This allows us to bound the r_o value of the wheel, which should also be maximized to create the greatest inertia for the wheel, as discussed above. Assume the housing diameter is between 5 in and 5.5 in lengthwise (along the direction of mounting), thus $r_o < 5$ in. Considering standard stock sizes helps narrow down values, and 4.75 in diameter stock was the largest available size less than 5 in. To account for tolerance in the size of the stock, set $r_o = \frac{4.7}{2} = 2.35$ in. With this value fixed, this brings the number of variables to three (r_i, h_1, h_2) , allowing for surface representation of the total inertia desired. Now solving for h_2 :

$$h_2 = \frac{\frac{2I_T}{\rho\pi} - r_o^4 h_1}{r_o^4 - r_i^4} \quad (2.11)$$

With r_o and I_T being known, that leaves h_2 dependent on r_i and h_1 , thus it is possible to visualize the acceptable combinations of these values as a surface. It is also possible to add a plane for total height h_T of the wheel (height of disk plus annulus). From the definition of h_1 and h_2 ,

$$h_T = h_1 + h_2 \quad (2.12)$$

this provides an additional plane which can be graphed in addition to the inertia surface to visually represent the values below the desired h_T . In this graph, $h_T = 1.5$ in was chosen to keep the total height of the wheel compact, while still allowing for

a large range of acceptable values. The variables r_i and h_1 were then restrained to a range of values and eq. (2.11) was used to generate the plot shown in fig. 2.5.

Inertia Sizing with Plane for $h_T = 1.5$

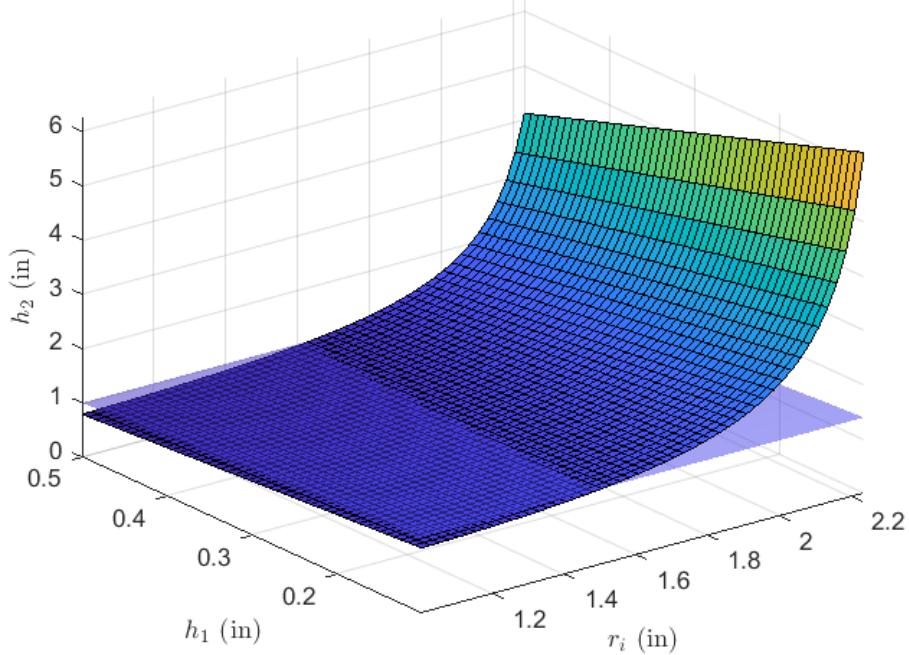


Figure 2.5: h_2 as a Function of r_o , I_T , and h_1

The wheel shape from fig. 2.4 and the associated inertia equations will serve as an approximation, as the actual wheel will have holes for screws, slight indents for component mating, and manufacturing tolerances that will cause differences from this derivation. To find a more accurate approximation of the mass properties of the wheel, it is possible to use CAD software. This will be further discussed in chapter 3.

Figure 2.5 gives a visualization of the acceptable values to reach the desired total inertia I_T , thus r_i , h_1 and h_2 values can be selected. As the wheel will be manufactured as a part of this work at Cal Poly SLO, it is advantageous to choose round values for machining purposes. From fig. 2.5, we want to choose an intersection of the the plane and the surface to reach the desired h_T of 1.5 in. It can be helpful to set a value of h_2 along this plane intersection and back out the values the associated r_i and

h_1 values. For h_2 , a value of 1.2 in was chosen to provide a mid-section height h_1 sufficient to support the load of the wheel (as shown in fig. 2.4) while keeping mass concentrated further from the axis of rotation. Basic FEA analysis confirmed that this general range provide sufficient load-bearing capabilities, however further FEA analysis would be beneficial for optimization of the wheel. This will be discussed in future work. Setting $h_2 = 1.2$ in results in fig. 2.6, giving us r_i , h_1 values to select from.

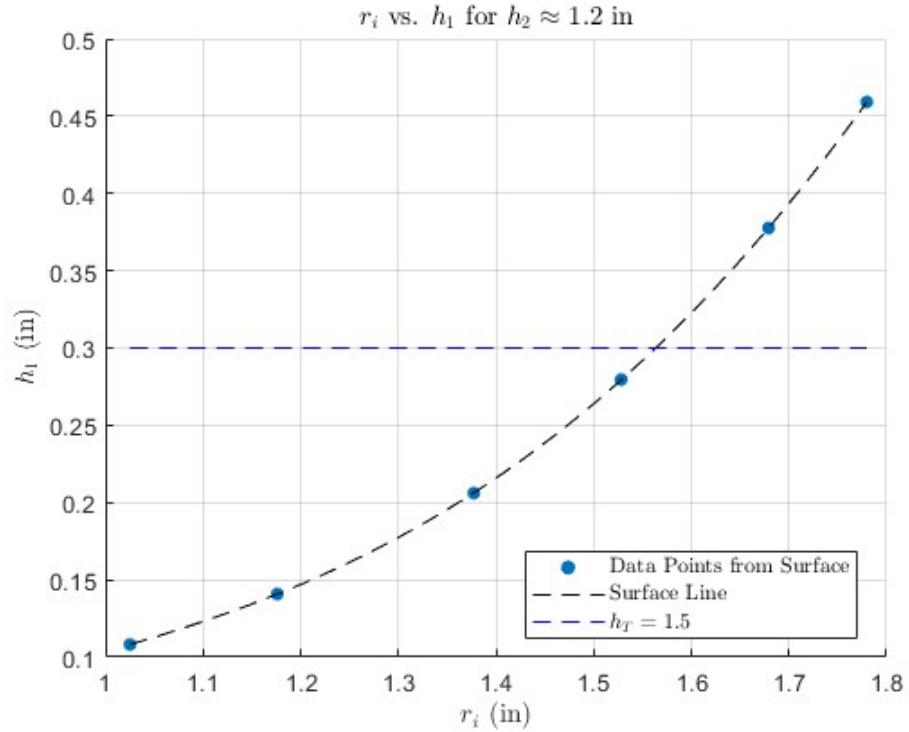


Figure 2.6: Fixed h_2 Plot from fig. 2.5

From fig. 2.6, values of $r_i = 1.6$ in and $h_1 = 0.3$ in were chosen to help create the initial sizing of the reaction wheel, as well as ensuring $h_T \leq 1.5$ in. Although these values are not exactly on the line from the surface, which represents the desired I_T , they provide round numbers to begin prototyping with, and the actual inertia of the wheel will be slightly different than this model due to an additional shaft and screws being a part of the spinning wheel. These chosen values will be reflected in the CAD

model of the wheel, discussed in chapter 3, which will also delve into the computer calculated inertia value.

2.3 Torque Requirements

Another important requirement is ensuring the reaction wheel and associated motor can actuate the system to the requirements of the system, in this case the SADS. In order to establish a method for ensuring this, some dynamics must be defined.

Angular acceleration is defined by:

$$\dot{\boldsymbol{\omega}} = \frac{d\boldsymbol{\omega}}{dt} \quad (2.13)$$

where $\boldsymbol{\omega}$ is the angular velocity, t is time. Torque is defined by:

$$\boldsymbol{\tau} = \mathbf{I}\dot{\boldsymbol{\omega}} \quad (2.14)$$

where $\boldsymbol{\tau}$ is the torque, \mathbf{I} is the inertia matrix, $\dot{\boldsymbol{\omega}}$ is the angular acceleration vector. eq. (2.14) is the case assuming that the angular velocity occurs strictly along one axis. When there is simultaneous rotation about multiple axes, dynamic coupling terms are introduced, resulting in Euler's equations of motion:

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) = \boldsymbol{\tau} \quad (2.15)$$

Expanding for each component, we obtain:

$$\begin{cases} I_x \dot{\omega}_x - (I_y - I_z)\omega_y\omega_z = \tau_x \\ I_y \dot{\omega}_y - (I_z - I_x)\omega_z\omega_x = \tau_y \\ I_z \dot{\omega}_z - (I_x - I_y)\omega_x\omega_y = \tau_z \end{cases} \quad (2.16)$$

For the purposes of sizing a single reaction wheel, it is assumed that the reaction wheel will rotate around a single axis and thus the single axis assumption of eq. (2.14) will be used. In reality, if there is any imperfect rotation about this single axis, the dynamics of the system would be better modeled by eq. (2.16).

These equations also hold for the SADS itself, and thus desired values from the SADS can levy requirements on the wheel capabilities. From eq. (2.14), torque is proportional to \mathbf{I} and $\dot{\boldsymbol{\omega}}$. The reaction wheels produced will have a maximum torque capability, and this torque must be sufficient to actuate the system as desired. Since inertia \mathbf{I} of the system is has already been determined for the SADS, this leaves the designer to choose a desired SADS actuation rate $\dot{\boldsymbol{\omega}}$ to determine the necessary torque to be produced by the reaction wheels.

For the SADS, the system is sized to simulate a small spacecraft and its dynamics, so it was appropriate to choose an $\dot{\boldsymbol{\omega}}$ derived from a hypothetical mission requirement. The requirement levied on the system was the system shall be able perform actuations at a rate necessary to track a ground object while in Low Earth Orbit (LEO). At a LEO altitude, the spacecraft would not match the Earth's rotational speed (as occurs in a Geostationary Orbit), so it would have to rotate relative to its orbital angular velocity to track a stationary point on the Earth's surface.

Although the torque requirement levied by this requirement ensures that the reaction wheels can perform the slew necessary to perform these maneuvers, it was not the driving requirement for sizing the wheels. Rather, it is meant as a check to ensure

that the sizing for the reaction wheels from momentum requirements could deliver the torque necessary to track a point on the Earth. The torque requirement for the system is set by similar industry capabilities, as previously discussed. This section will establish that this torque capability meets the requirements necessary for the ground tracking scenario posed. The methodology for computing the required torque to meet this actuation rate is as follows.

From eq. (2.14), it is clear that a preliminary estimate of the inertia \mathbf{I} of the system is essential for providing a range for motor sizing. The preliminary estimate of the inertia of the SADS was provided by previous work by Gilman [11] as:

$$\mathbf{I} = \begin{bmatrix} 2.21 & .006 & .006 \\ .006 & 1.701 & .008 \\ .006 & .008 & 1.798 \end{bmatrix} \text{ kg} \cdot \text{m}^2 \quad (2.17)$$

However, due to ongoing work on the balancing platform and the addition of other subsystems, the final inertia of the system will change. The inertia of the platform should still be similar to the initial estimate, providing a satisfactory reference for performing calculations throughout this section.

For the SADS, simulating a small satellite is of interest, thus it is necessary to set a relevant orbit. An altitude h of 300 km was set as the desired orbit altitude for the spacecraft, as this altitude is on the lower range LEO for small satellites and thus requires higher actuation capabilities. For simplification purposes, this will assume planar motion between the rotation of the Earth and the rotation of the spacecraft, as well a prograde orbit. The specific case that will be considered is a satellite in an equatorial orbit (inclination of 0°) tracking a point on the Earth's equator as seen in fig. 2.7 for simplification purposes. A more complex orbit could also be utilized, but

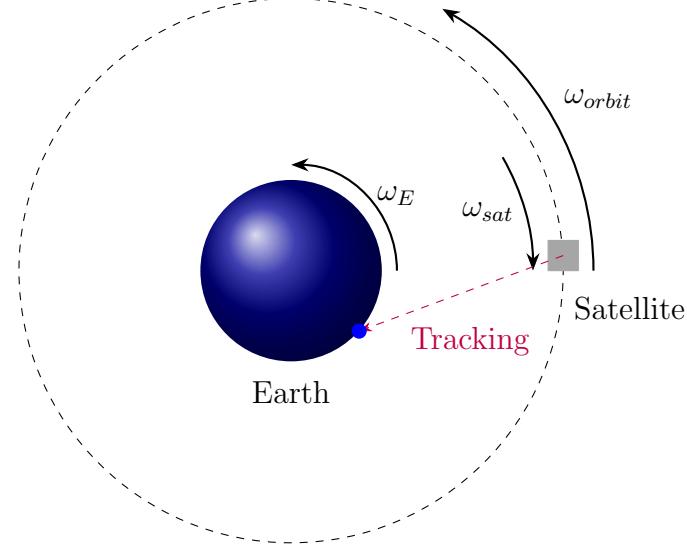


Figure 2.7: Satellite in LEO Tracking a Ground Point

this provides a general range for LEO requirements. To find the required slew rate of the spacecraft to track a ground object as described, start by calculating the velocity of the spacecraft and associated angular velocity.

$$r = r_E + h \quad (2.18)$$

$$v = \sqrt{\frac{\mu}{r}} \quad (2.19)$$

$$\omega_{orbit} = \frac{v}{r} \quad (2.20)$$

Earth's rotation ω_E must also be taken into account for tracking a ground system ($\omega_E = 7.29 \times 10^{-5}$ rad/s). To find the relative angular velocity between the Earth and the spacecraft, use the following equation:

$$\omega_{sat} = \omega_{orbit} - \omega_E = 1.16 \times 10^{-3} - 7.29 \times 10^{-5} = 1.08 \times 10^{-3} \text{ rad/s} \quad (2.21)$$

Making a planar assumption results in a three-dimensional angular velocity with zero components excluding the direction of rotation. Define the angular velocity vector with rotation purely in the \hat{y} -direction (this is discussed later in the section and visualized in fig. 2.8) as:

$$\boldsymbol{\omega} = \begin{bmatrix} 0 \\ \omega_{sat} \\ 0 \end{bmatrix} \quad (2.22)$$

This allows us to begin the process of calculating the required torques to actuate this rate.

Assuming constant acceleration to reach this rate from rest, it is necessary to set a time requirement t_{slew} to reach this angular velocity. For this process, 15 seconds was chosen. Thus, angular acceleration required $\dot{\omega}_{sat}$ is defined as follows:

$$\dot{\omega}_{sat} = \frac{\omega_{sat}}{t_{slew}} = \frac{1.08 \times 10^{-3}}{15} = 7.23 \times 10^{-5} \text{ rad/s}^2 \quad (2.23)$$

From equation eq. (2.14), the torque required on the system is $\boldsymbol{\tau}_{req} = [0.04 \times 10^{-4}, 1.23 \times 10^{-4}, 0.06 \times 10^{-4}]^T$ assuming perfect rotation around the y-axis with no gyroscopic effects. To add a margin of safety, set double this torque as the maximum required torque, $\boldsymbol{\tau}_{max} = [0.09 \times 10^{-4}, 2.46 \times 10^{-4}, 0.12 \times 10^{-4}]^T$. This provides the torque in terms of the system, but we are curious how to obtain the torque required

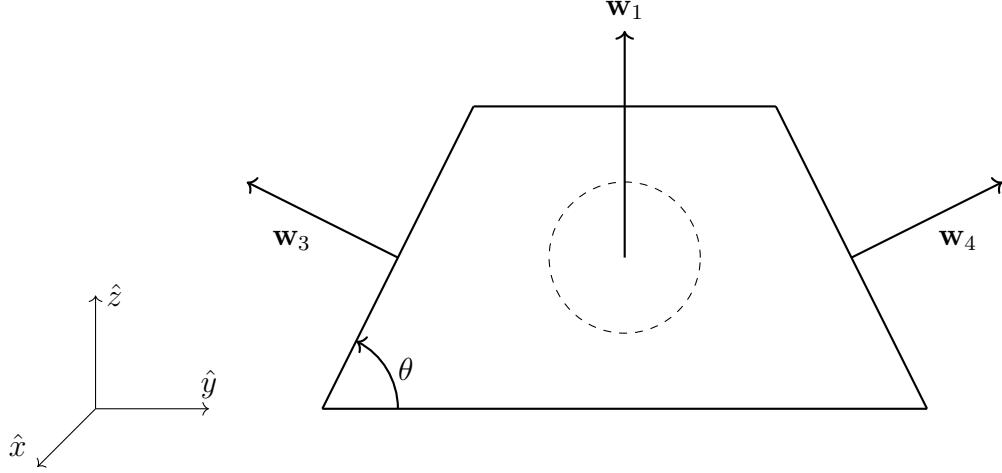


Figure 2.8: Updated Axis for Pyramidal Reaction Wheel System

from the individual reaction wheels. It is necessary to develop a method for mapping the torque produced from the four reaction wheels onto the three axis spacecraft.

As previously mentioned, the SADS uses a pyramidal configuration of reaction wheels, as shown in fig. 1.2. The body-fixed reference frames are denoted $\mathbf{e}_1, \mathbf{e}_2$, and \mathbf{e}_3 . The spin axis of the four reaction wheels is denoted by $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$, and, \mathbf{w}_4 . Throughout this thesis, a differently defined $\hat{x}, \hat{y}, \hat{z}$ body axis system will be used and referenced as compared to the $\mathbf{e}_1, \mathbf{e}_2$, and \mathbf{e}_3 of fig. 1.2. The updated axis definition aligns the \hat{x} axis with the planar direction of \mathbf{w}_1 in the \hat{x}, \hat{y} plane, the \hat{y} axis with the planar direction of \mathbf{w}_4 in the \hat{x}, \hat{y} plane, and the \hat{z} axis completes the coordinate system through the cross product.

The new axes are illustrated in fig. 2.8. The spin axes of the reaction wheels are still denoted by \mathbf{w} , and θ represents the mounting angle of the wheel.

The transformation matrix from the reaction wheel coordinate system to the body frame coordinate system from fig. 2.8 is given by:

$$\mathcal{W}_4 = \begin{bmatrix} a & -a & 0 & 0 \\ 0 & 0 & c & -c \\ b & b & d & d \end{bmatrix} \quad (2.24)$$

where $a = c = \cos(\theta)$, $b = d = \sin(\theta)$, thus $a^2 + b^2 = c^2 + d^2 = 1$. This matrix assumes a preferred direction along the body \hat{z} axis, with each reaction wheel contributing to this alignment. When $a = c$ and $b = d$, the spin axes form the same angle θ with the \hat{x}, \hat{y} plane. The mounting angle of the reaction wheel θ is 63° from the existing SADS platform. This configuration is the most commonly used for Earth-pointing spacecraft, where maintaining a constant orientation along the \hat{z} axis is necessary to keep instruments oriented toward the Earth's surface.

As shown earlier, the known \mathbf{I} and $\dot{\omega}$ yields the torque vector $\boldsymbol{\tau}_{req} = [\tau_x, \tau_y, \tau_z]^T = [0.04 \times 10^{-4}, 1.23 \times 10^{-4}, 0.06 \times 10^{-4}]^T$ that needs to be generated by the reaction wheels. In order to determine the torques required by each of the four reaction wheels, the reaction wheel torques must be mapped to the body frame by:

$$\boldsymbol{\tau}_{req} = \mathcal{W}_4 \cdot \boldsymbol{\tau}_w \quad (2.25)$$

where $\boldsymbol{\tau}_w = [\tau_1, \tau_2, \tau_3, \tau_4]^T$ is the vector of torques for each of the four reaction wheels. It is possible to solve for $\boldsymbol{\tau}_w$ by using psuedoinverse matrix multiplication. The psuedoinverse is required due to the fact that \mathcal{W}_4 is not a square matrix. The psuedoinverse for a given matrix A is defined by:

$$A^+ = (A^T A)^{-1} A^T \quad (2.26)$$

Using the psuedoinverse of \mathcal{W}_4 yields:

$$\boldsymbol{\tau}_w = \mathcal{W}_4^+ \cdot \boldsymbol{\tau} \quad (2.27)$$

Performing all of these calculations, the following torque vector is generated for all four wheels in the case of $\boldsymbol{\tau}_{max}$:

$$\boldsymbol{\tau}_w = \begin{bmatrix} \tau_{w_1} \\ \tau_{w_2} \\ \tau_{w_3} \\ \tau_{w_4} \end{bmatrix} = \begin{bmatrix} 1.28 \times 10^{-4} \\ -0.63 \times 10^{-4} \\ 2.71 \times 10^{-4} \\ -2.70 \times 10^{-4} \end{bmatrix} \text{ N m} \quad (2.28)$$

From eq. (2.28), the maximum torque component required to slew the platform at the rate described by $\boldsymbol{\tau}_{max}$ is 2.71×10^{-4} N m. As seen in table 2.2, the selected motor easily fulfills this requirement with a rated torque of 25.4×10^{-3} N m. This larger torque capability will also allow us to perform faster maneuvers than the rate required by the ground tracking scenario.

2.4 Design Requirements Conclusion

To conclude, a list of the desired reaction wheel design specifications has been provided in table 2.3. The values in this table will serve as a point of reference when the actual wheel is assembled and tested, to determine how close the actual wheel was to the design requirements. The wheel inertia speed is slightly different than the one derived in this section, and this is due to this being the CAD model inertia, which will be discussed in the following chapter. This table will be updated in chapter 4 with the

tested values as well as the percent difference between those values and the design specifications.

Table 2.3: Key Reaction Wheel Design Specifications

Reaction Wheel Design Specifications	
Momentum Capacity	0.500 N m s
Wheel Inertia	1.698×10^{-3} kg m ²
Max Wheel Speed	2760 rpm
SADS Angular Acceleration	7.23×10^{-5} rad/s ²

Chapter 3

HARDWARE AND SYSTEM DEVELOPMENT

After determining the required characteristics of the wheel and motor, as well as selecting available hardware, the next stage was to design a system that integrated all of the components into a single system. This section will delve into the design of the reaction wheel assembly, as well as how the assembly functions and integrates with a larger system such as the SADS.

3.1 Design

As with many systems, when performing the initial design of this system, there were uncertainties in how the assembly would function, be manufactured, and assemble. This led to several design iterations for individual parts and consideration of multiple options. An initial emphasis on the first design iteration was the ability to produce the assembly entirely at Cal Poly SLO. The initial designs were then constructed so that parts could be made by the tools available to students at Cal Poly such hand-operated lathes, mills, etc. Inevitably, this led to imperfections in manufacturing, which causes issues especially in a high-precision system like a reaction wheel. Thus, the design emphasis was shifted to utilizing CNC manufacturing to achieve desired accuracy. This influenced design choices, and also allowed for greater freedoms in the design process.

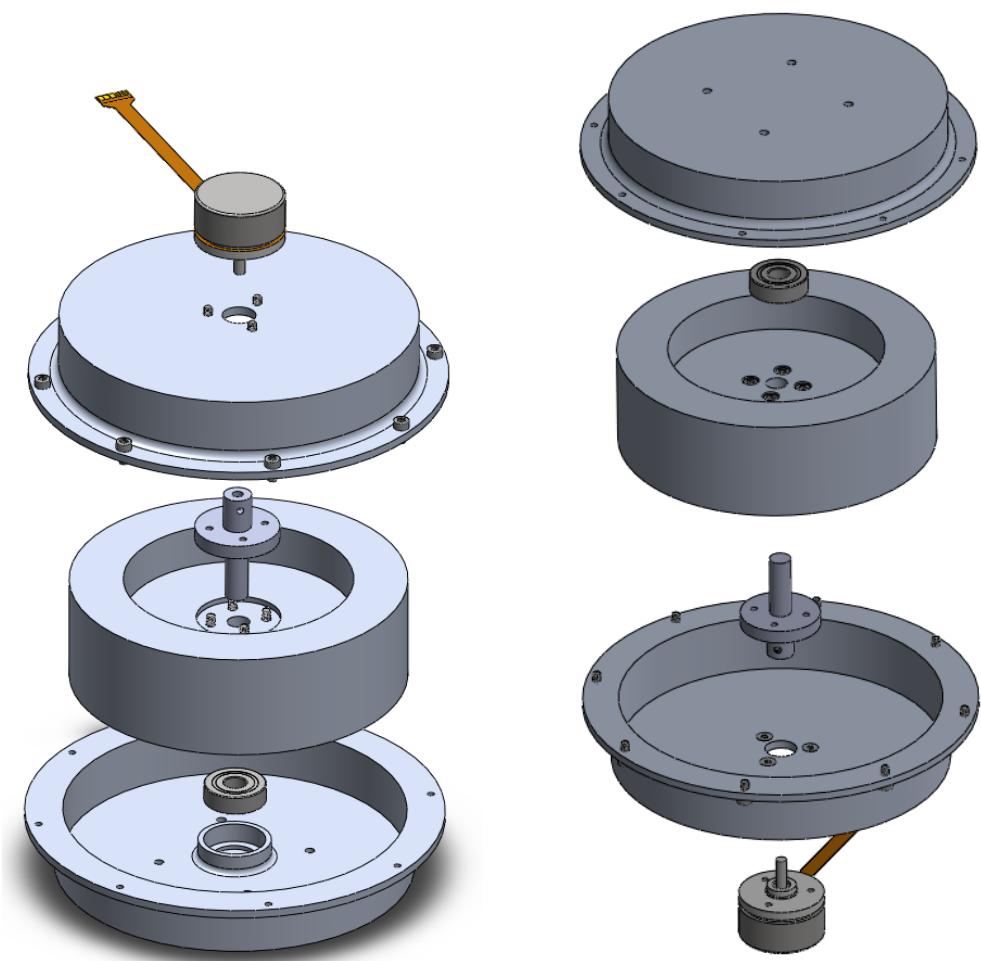
The first step in the design process was to use the requirements and components developed in chapter 2. Factors such as having a preselected motor and sizing of the wheel itself were the foundations upon which the rest of the design was created. The

rest of the components required to complete the assembly were designed to ensure functionality with these components.

The most important part of the assembly is the reaction wheel itself. The requirements for the reaction wheel were developed in section 2.2.3. The important geometric values for developing an initial reaction wheel were a total height $h_T = 1.5$ in, solid disk height $h_1 = 0.3$ in, annulus height without the solid disk $h_2 = 1.2$ in, outer radius $r_o = 2.35$ in, and inner radius $r_i = 1.6$ in.

From this, the general dimensions of the wheel were known. In order to spin the wheel, it was necessary to mate the motor shaft and the wheel, and an intermediate shaft was designed to do so. This intermediate shaft used a set screw to mate to the motor shaft, and is attached by four screws to the wheel itself. In addition, this shaft also extends past the mating with the wheel to sit in a bearing, to ensure axial rotation of the assembly, and reduce axial translation as pictured in fig. 1.9. The bearing is held in the housing, which itself is split into two halves that are joined by eight screws. The housing also holds the motor, as well as has holes drilled into it for mounting to other systems (such as the SADS). This reaction wheel assembly is pictured in fig. 3.1 and fig. 3.2. Both an exploded view of the individual components and an assembled view are provided for clarity.

Figure 3.3 gives insight into the inside of the reaction wheel assembly fully assembled. The cross section of the wheel is very similar to that of fig. 2.4. Additionally, there is empty space between the wheel and the housing, which allows for screws to be used there to mount the reaction wheel assembly to an outside system. The shaft sits slightly above the housing level, ensuring no contact, but still remains within the bearing. The shaft also sits in an indent within the wheel itself, allowing for better mating between the two parts.



(a) Top Angled View

(b) Bottom Angled View

Figure 3.1: Exploded Views of Reaction Wheel Assembly

With the CAD model, it was to calculate the inertia value rotating assembly. The inertia matrix for the rotating part of the assembly (wheel, shaft, and screws) is:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} = \begin{bmatrix} 9.248 \times 10^{-4} & 0 & 0 \\ 0 & 1.698 \times 10^{-3} & 0 \\ 0 & 0 & 9.248 \times 10^{-4} \end{bmatrix} [\text{kg}\cdot\text{m}^2]$$

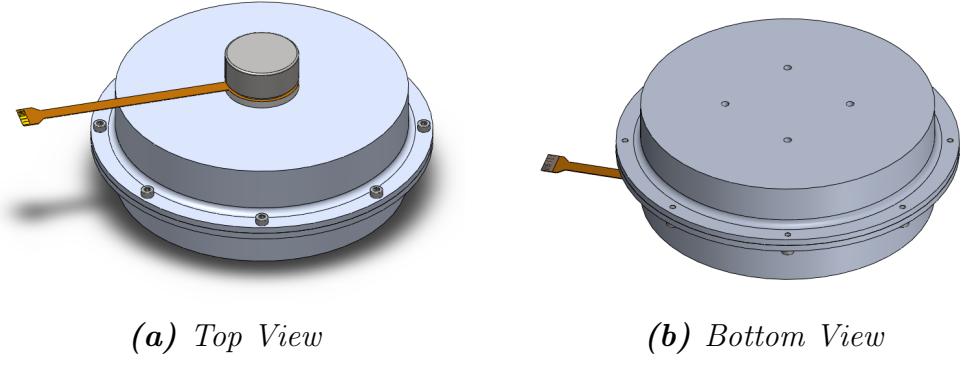


Figure 3.2: Assembled Views of Reaction Wheel Assembly

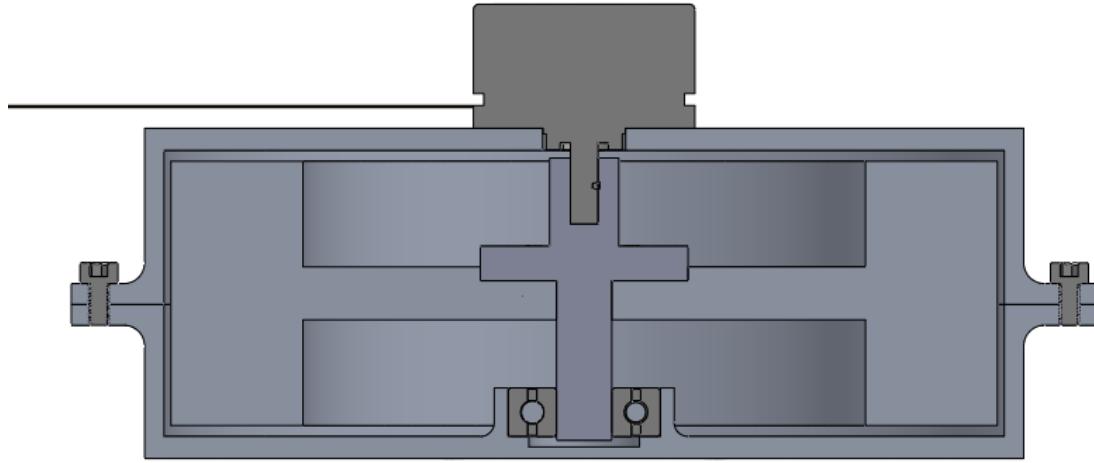


Figure 3.3: Sliced View of Reaction Wheel Assembly

with $I_{yy} = 1.698 \times 10^{-3} \text{ kg m}^2$ being the inertia about the spin axis. This is relatively close to the previously calculated required Inertia value $I_{wheel} = 1.73 \times 10^{-3} \text{ kg m}^2$, demonstrating that the design satisfies the inertia requirement of the system.

As previously mentioned in chapter 2, the selected motor for the reaction wheel system was a Nanotec DF32M024027-A - Brushless DC flat motor. In order to interface and control the motor, two additional components were required: a Nanotec CL3-E-1-OF Motor controller [5] and a ZIB-DF32 additional board for connecting to the FPC cable of the motor, pictured in fig. 3.4. These are not included in the CAD model, but

will be a part of the final assembly. Due to on-going testing, these components were left free standing out of the assembling for ease of access modifying connections and programming. The motor was also left external in the design for prototype testing and braking capabilities.

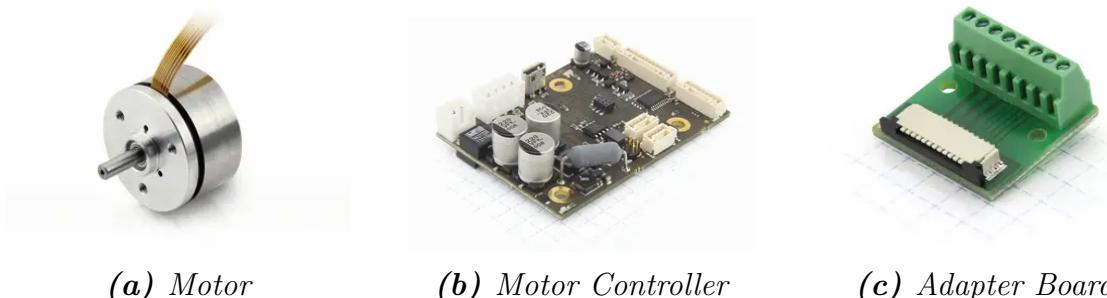


Figure 3.4: Required Electronic Components [5]

With all the parts designed and shown, this allows us to create a bill of materials, listed in table 3.1, for the reaction wheel assembly. Individual components such as the wheel and shaft are not available commercially, so the corresponding stock required to machine them will be listed. The length of the stock is based upon the length of the parts in the assembly, but may not be recommended for manufacturing purposes or commercially available at that length.

Table 3.1: Bill of Materials

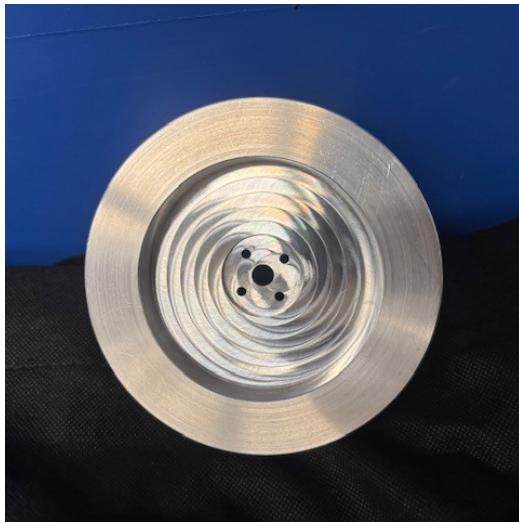
Description	Quantity/Length
DF32M024027-A - Brushless DC flat motor	1
CL3-E-1-0F Motor controller	1
ZIB-DF32 Additional Board	1
1.25 in Diameter AL6061 stock	1.6 in
4.75 in Diameter AL6061 stock	1.5 in
6 in Diameter AL6061 stock	2 in
6 in Diameter AL6061 stock	2 in
8 mm Inner Diameter Bearing	1
M3 Counterbore Screw (6mm Length)	3
M3 Counterbore Screw (6mm Length)	12
M3 Set Screw (4mm Length)	1

3.2 Manufacturing

The components required to build the assembly required manufacturing, and it was chosen to CNC (computer numerical control) mill/lathe the parts in addition to operations done by hand such as tapping and drilling. The housing for the assembly was sent to an external party for manufacture, but the wheel and shaft itself were manufactured using Cal Poly facilities. The components were designed for manufacturing and were able to be produced by students at Cal Poly. Additionally, the components were designed so that all that was required to assemble them were hex keys. The manufactured components are shown in fig. 3.5.

Although the components were manufactured using CNC operations, there was still some imperfections in the final products. This is crucial in our application, as slight imbalances in the system can result in disturbances that decrease the performance of our system, as discussed in chapter 1. The imperfections were on the order of thousands of an inch, so small in error but still holding some significance. Due to the imperfections, it would be beneficial to balance the system using testing techniques to find where the imbalances within the wheel are and add/subtract mass to adjust the balance accordingly. Techniques and testing for this were discussed in chapter 1. However, this will lie outside the scope of this thesis, and will be included in the future work section.

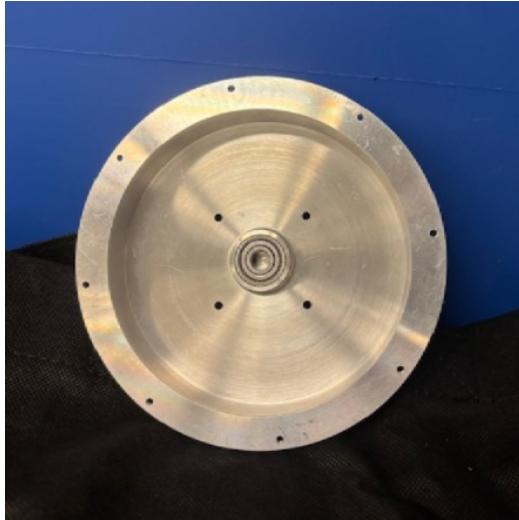
After manufacturing the final assembly came together as designed, pictured in fig. 3.6. The motor is attached, but the additional electronics (motor controller and cable adapter board) are not included in this picture.



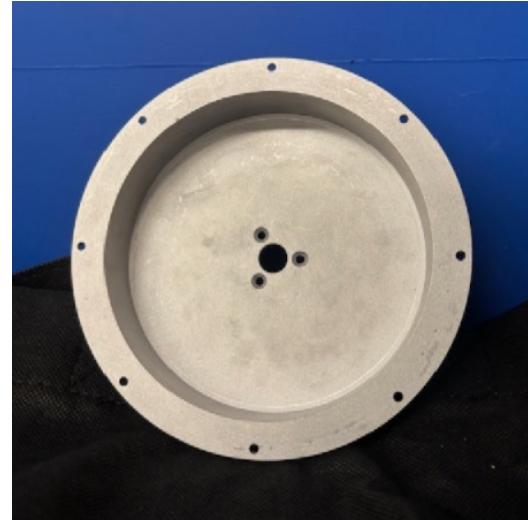
(a) Wheel



(b) Shaft



(c) Housing Bearing Side



(d) Housing Motor Side

Figure 3.5: Manufactured Assembly Components

3.3 Electronics

With the manufacturing completed and the reaction wheel assembled, the next step was setting up the electronics.

Both the motor and motor controller required a 24 V power source, thus a 24 V battery sufficed for powering the system. The motor had a 0.5 A rated current, but

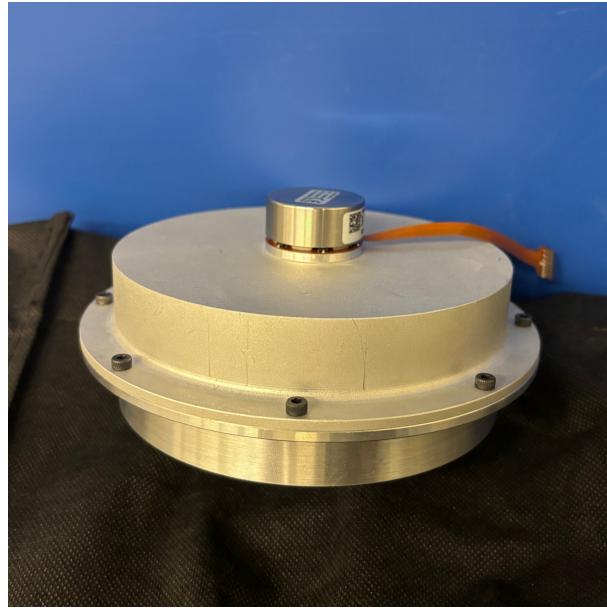


Figure 3.6: Manufactured Assembly

drew less current than that during most operation. The motor controller also has a 10 V output, which was used to power the Arduino Uno Rev3. The Arduino powered the MPU-9250 as seen in fig. 3.7 (a 9-axis sensor with accelerometer, gyroscope, and magnetometer), and received position data from it. Additionally, an op-amp was used (application discussed below) to amplify a signal up to 10 V, so it was wired in parallel with the 24 V source to have sufficient voltage to be able to amplify to the 10 V required. Thus, all the components in the system were powered by the 24 V battery.



(a) *MPU9250 Attitude Sensor*



(b) *MCP2515 CAN Module*

Figure 3.7: Additional Electronic Components

For basic control, there were several approaches for initial testing and set up. Nanotec offers a software called Plug and Drive Studio 3, which is a quick method of establishing basic motor function. From this interface, it is possible to control torque, velocity, position, and configure the motor in set values and durations. The specific commands sent via Plug and Drive Studio 3 will be further discussed in chapter 4.

The motor controller also accepted an analog input of 0-10 V that could be read and used within the code on the controller, which was utilized to control the output torque. This allowed external sources to determine the performance of the motor. Two methods were used to generate an analog signal for torque control. First, was a potentiometer allowing the operator to vary a voltage by hand, generating an analog signal from 0-10 V. The second method utilized the analog output capabilities of the Arduino via PWM (pulse width modulation) pins, which allowed the analog signal to be varied by code on the Arduino. The Arduino was only capable of generating a maximum of 5 V output, thus in order to retain full control capabilities via the 0-10 V analog input to the motor controller, it was necessary to use an op-amp. A non-inverting op-amp configuration was used, as shown in fig. 3.8. The gain from a non-inverting op-amp is:

$$\frac{V_{out}}{V_{in}} = 1 + \frac{R_2}{R_1} \quad (3.1)$$

Because the Arduino's maximum output is 5 V (V_{in}), a gain of 2 is desired to reach 10 V (V_{out}) for the motor controller. Setting $R_1 = R_2$ results in a gain of 2, allowing for the Arduino to have full control capability of the motor via an analog input.

With the Arduino control capability, it was possible to use the MPU-9250 position data and a control law to generate torque commands for the wheel and achieve basic control capabilities. This will be further discussed in the following chapter, as well

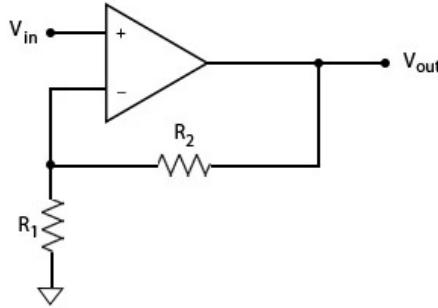


Figure 3.8: Non-Inverting Operational Amplifier [19]

as additional tests to demonstrate the capabilities of the wheel. To give a basic illustration of the architecture of the single reaction wheel system for testing, all of these components are shown in fig. 3.9. This diagram shows the connections between various parts, as well as the types of connections between them. Ultimately, the motor is connected to the reaction wheel which leads to its control. In order to collect data from the motor as presented in chapter 4, an SD card module was utilized in parallel with the CAN module, and was able to read values from the motor's hall sensors. Finally, a switch was added to enable the motor controller to switch off commands from the motor controller to the motor as well, providing a way to safely stop wheel function.

The current configuration has CANopen capability via a MCP2515 CAN Bus Module, as seen in fig. 3.7, that interfaces with the Serial Peripheral Interface, or SPI, on the Arduino. This CANopen was used to communicate directly with the motor, and request velocity and torque values from the hall sensors as previously discussed. The usage of CANopen was limited to simple data requests for the primary characterization of the wheel, however it is possible to utilize this for control in future configurations.

Single Reaction Wheel Test Configuration

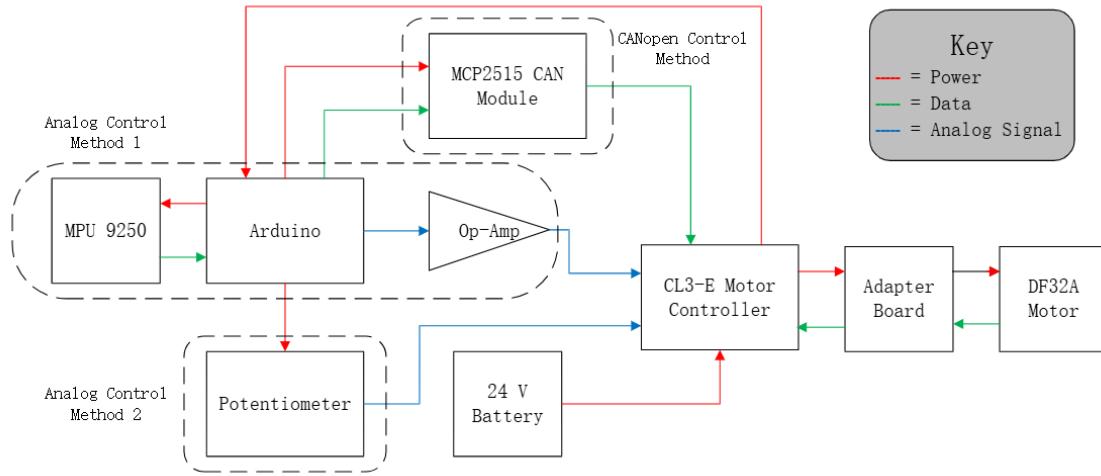


Figure 3.9: Single Reaction Wheel Electronics Configuration for Testing

For combining multiple reaction wheels and thus multiple motors and motor controllers, utilizing a CANopen bus between a flight computer and the reaction wheels would be a efficient, simple way to establish communication. A hypothetical CANopen configuration for the SADS with four reaction wheels is illustrated in fig. 3.10. The motor controller is designed for usage with CANopen, and has a designated port for CANopen communication. This is especially beneficial for the SADS as a whole, which will have one central flight computer. Future work will lead to a pyramidal reaction wheel configuration with four wheels as discussed in chapter 1 for the SADS with a flight computer used to control all of the wheels. This flight computer would be able to use CANopen to communicate simultaneously with the reaction wheels, simplifying the communication structure between the attitude control system and the flight computer as seen in fig. 1.7. There would also be no need for an intermediate microcontroller (Arduino), as the microcontroller in the current configuration reads IMU data and generates an analog signal which is then used to control the reaction wheel. The reading of IMU data would be handled by the flight computer, and then

could generate a control command to be actuated via CANopen communication to the motor controller.

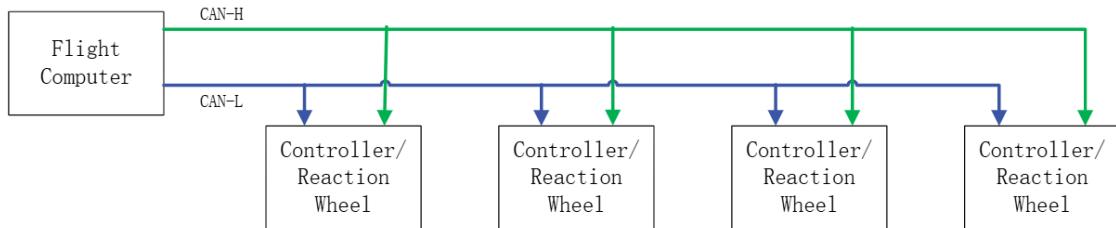


Figure 3.10: CANopen Possible Configuration for SADS

The motor controller also offers compatibility with MODBUS RTU for communication, however CANopen was determined to be the better communication protocol for this system. CANopen offered real-time, priority-based messaging as well as error detection and fault reporting, all of which are beneficial to the reaction wheel system.

Chapter 4

SOFTWARE AND TESTING

This chapter will introduce the testing methods that were implemented with the single reaction wheel system. Due to time constraints and limited testing equipment, a full-fledged reaction wheel characterization was not able to be performed, such as testing of static/dynamic imbalance, wheel speed and vibration characterization, and friction. The testing conducted in this thesis validates the design parameters and demonstrates control capability of the reaction wheel. However, further testing would be beneficial and will be discussed in future work.

The main goal of these tests is to determine how accurately the reaction wheel performs as compared to its design parameters. There are inherently imperfections in the methods utilized to perform these tests, however they provide an elementary method of determining the desired characteristics of the wheel. Potential sources of error will be discussed later in this chapter.

4.1 Software

In order to send commands to control the motor, it was necessary to establish software that could perform the necessary tasks. As previously mentioned, Plug and Drive Studio 3 is an application developed by Nanotec for basic control and commissioning of the motor. Basic control capabilities included performing position, velocity, and torque motion tests. Commissioning the motor ensured proper function and data readout. Motor controller code development could be achieved within this software as well, as the motor controller was able to be programmed via NanoJ, a programming

language similar to C/C++ and developed by Nanotec for use with its products. NanoJ can send torque, velocity, position values and more the motor, as well as read inputs into the controller.

As mentioned in section 3.3, analog input to the motor controller was used for testing outside of Plug and Drive Studio 3. For the application of a reaction wheel, commanding torque is of the most interest, thus the analog signal was used to generate a torque command τ to the motor via NanoJ. The NanoJ code developed for the motor controller translated this analog input into the motor torque and is included in appendix B.

With the op-amp set-up previously discussed, the analog signal could be varied by the Arduino from 0-10 V, which could then be mapped to the torque output. The motor was capable of producing both positive and negative torque, so it was necessary to develop a voltage to torque mapping scheme that was able to achieve max torque $\pm\tau_{max}$. In order to maintain 0 V as zero-output torque and maintaining 10 V of range for controlling either torque, a digital pin was used introduced to interpret the voltage as positive or negative, due to the Arduino being incapable of directly producing a negative voltage. If a negative torque was required, the digital pin output was set to LOW, and conversely HIGH for positive torque, as illustrated in fig. 4.1. The motor controller was programmed to interpret the combination of the analog signal and the digital pin accordingly and produce the required torque output. As previously mentioned, the motor controller only operated if the value provided by the switch was HIGH as well. If the switch value was LOW, the output torque was set to zero.

The motor controller sampled the digital/analog inputs once per millisecond (1000 Hz), and signal changes less than one millisecond in duration were not processed. Additionally, the NanoJ code on the controller received computing time cyclically in a 1 millisecond clock, which aligns with the sampling of the inputs. The relationship

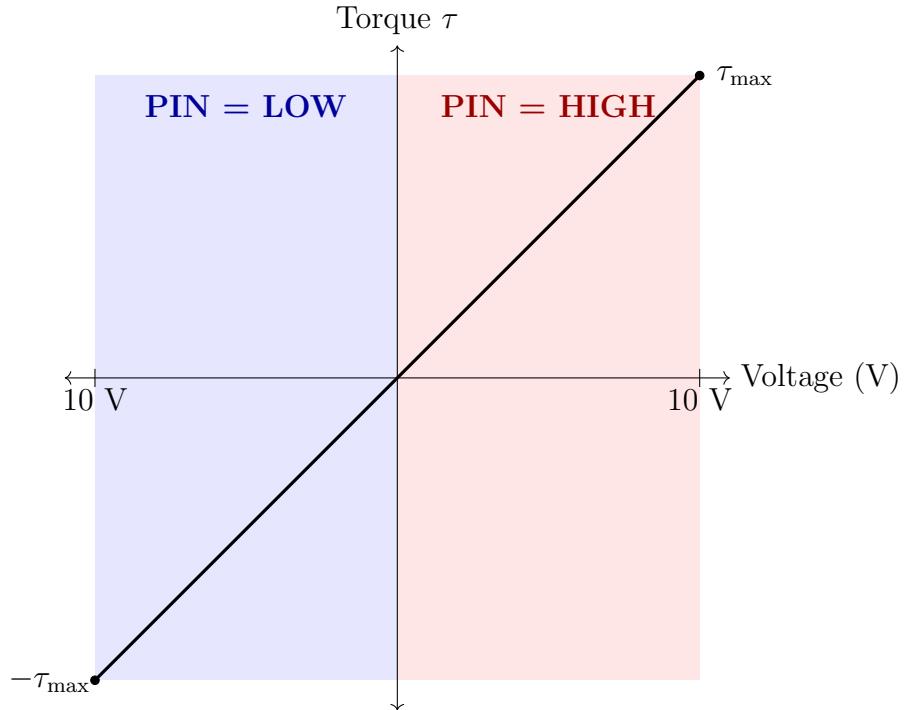


Figure 4.1: Voltage to Torque Diagram

is shown in fig. 4.2. The IMU was also able to be sampled at 1000 Hz, thus the maximum frequency able to be achieved in code for both the motor controller and the Arduino operated was 1000 Hz. For control maneuvers however, this high of a frequency is not necessarily required.

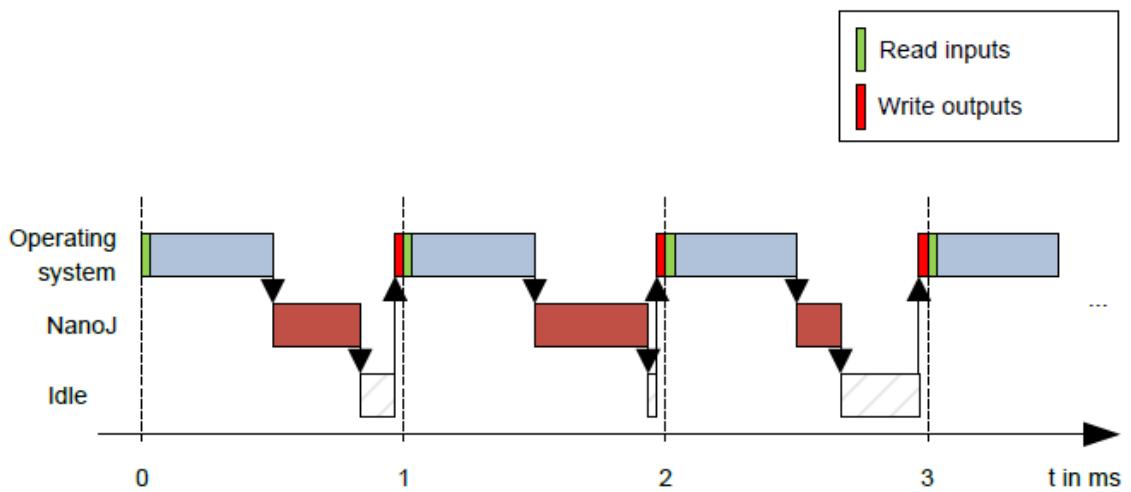


Figure 4.2: NanoJ Operating Frequency [5]

With the architecture set up for how the motor controller interpreted an analog signal, various software configurations could be tested. These tests are discussed below.

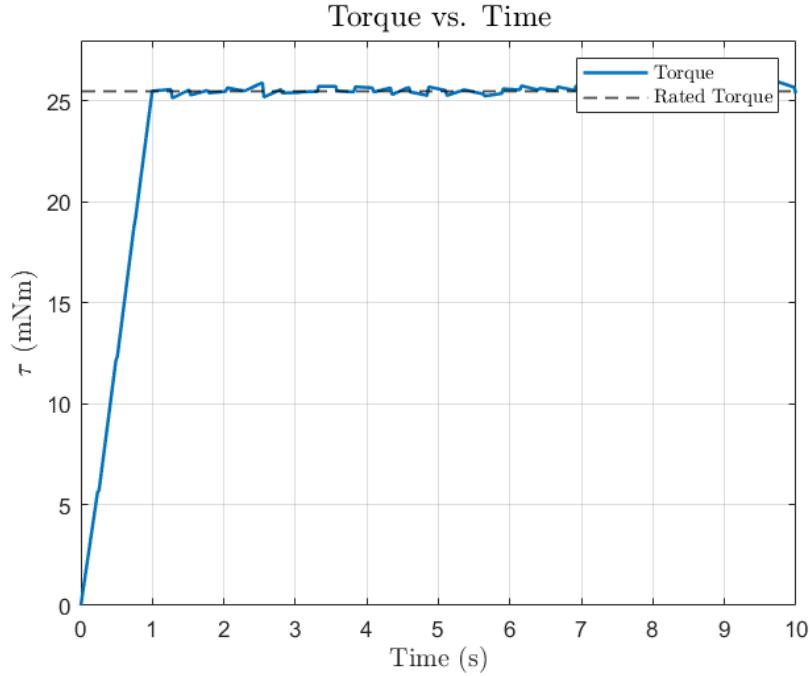
4.2 Validating Design Specifications

As discussed in chapter 2, the wheel was designed to meet certain specifications. The four most important were desired inertia I_{wheel} , angular momentum H_{wheel} , max wheel speed ω_{wheel} , and SADS angular acceleration $\dot{\omega}_{SADS}$. The design values were wheel inertia $I_{wheel} = 1.698 \times 10^{-3} \text{ kg m}^2$, wheel momentum $H_{wheel} = 0.500 \text{ N m s}$, wheel angular speed $\omega_{wheel} = 2760 \text{ rpm}$, and SADS angular acceleration $\dot{\omega}_{SADS} = 7.23 \times 10^{-5} \text{ rad/s}^2$.

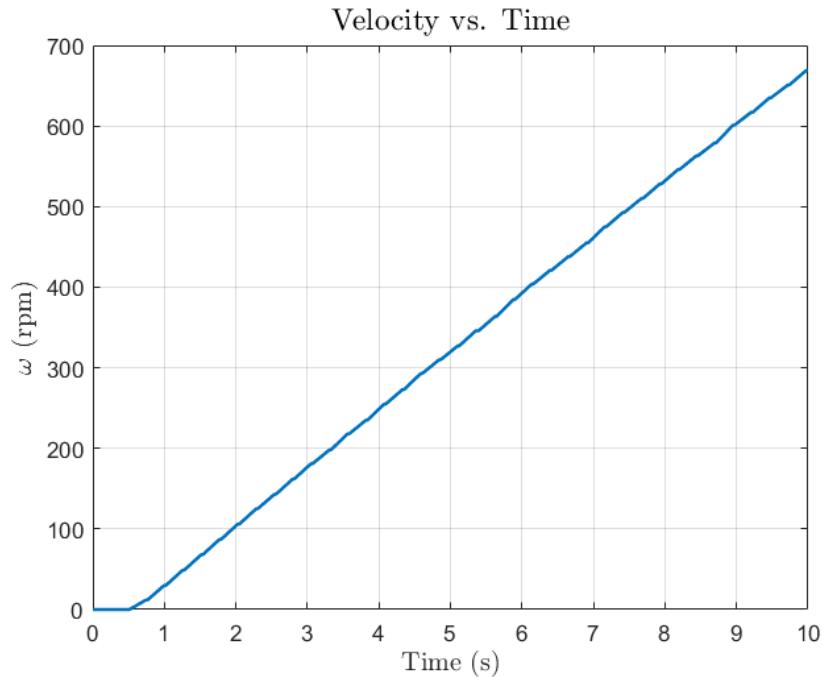
In order to determine these values, the reaction wheel was tested in both a standalone configuration and mounted on the SADS. For the purpose of determining these values, a combination of Plug and Drive Studio 3 and analog inputs were used to control the wheel.

4.2.1 Testing Inertia of the Wheel

To find the inertia of the wheel, a known torque was input into the wheel, and a resulting angular acceleration was measured to determine inertia, as shown by the relationship in eq. (2.14). The known torque inputted was the rated torque of the motor, and the angular acceleration was calculated by performing a linear regression on the slope of the angular velocity versus time. The test results are shown in fig. 4.3. The values used to find the inertia were taken after the rated torque and a constant acceleration had been reached.



(a) Torque for Angular Acceleration



(b) Angular Acceleration

Figure 4.3: Plots for Determining Inertia

From this test, the inertia of the wheel I_{wheel} was determined to be 1.7164×10^{-3} kg m², yielding an error of 1.083 % compared the expected design value from CAD of $I_{wheel} = 1.698 \times 10^{-3}$.

Also, by zooming in on the beginning of the velocity versus time plot, it is possible to observe the effects of static friction on the system. As shown in fig. 4.4, there is a brief period of time, approximately 0.5 seconds where there is no motion, even as the torque has begun in the system. Clearly, static friction is present, and would effect the behavior of the system and lead to errors in calculations. Characterizing both the static and dynamic friction present in the system will be included in the future work part of the document.

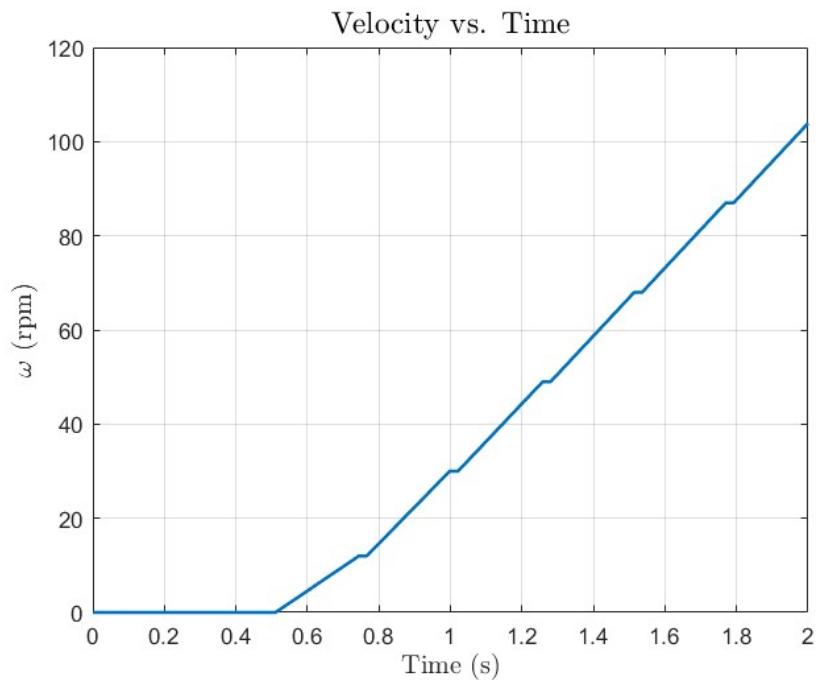


Figure 4.4: Zoomed In View of Figure 4.3b to Show Effects of Static Friction

4.2.2 Testing Momentum Capacity of the Wheel Mounted on SADS

With the inertia of the wheel validated, it was possible to determine the angular momentum of the wheel using eq. (2.2). The maximum speed of the wheel ω_{wheel} was determined by executing a full rated torque of the motor and allowing the wheel speed to reach a maximum. For determining the maximum speed, it was decided to mount the single wheel on the SADS to observe its behavior on the system it was designed

for. In addition, the resulting motion of the SADS platform could be recorded by the IMU on the SADS and verify that the wheel was capable of performing actuations on the SADS. A 3D printed mount was used to attach the reaction wheel to the SADS, as pictured in fig. 4.5, however for future mounting purposes a rigid, vibration resistant mount made of a higher quality material should be manufactured. Additionally, the electronics would be encapsulated on the reaction wheel itself as previously discussed, as in this test they were simply put onto the SADS platform itself. The system was powered off a 24V battery for this test.

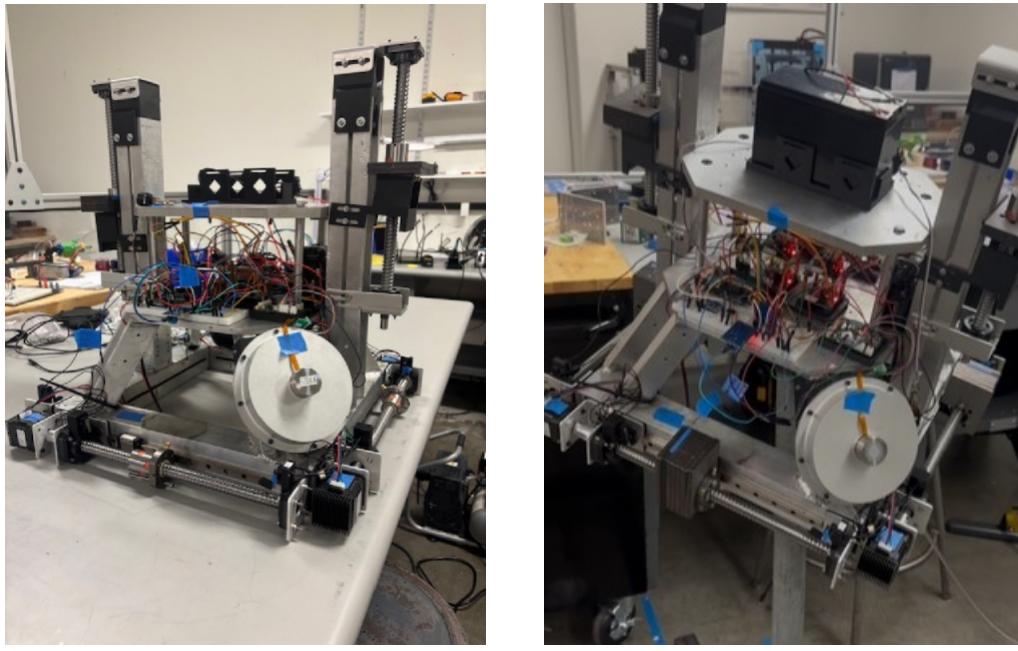


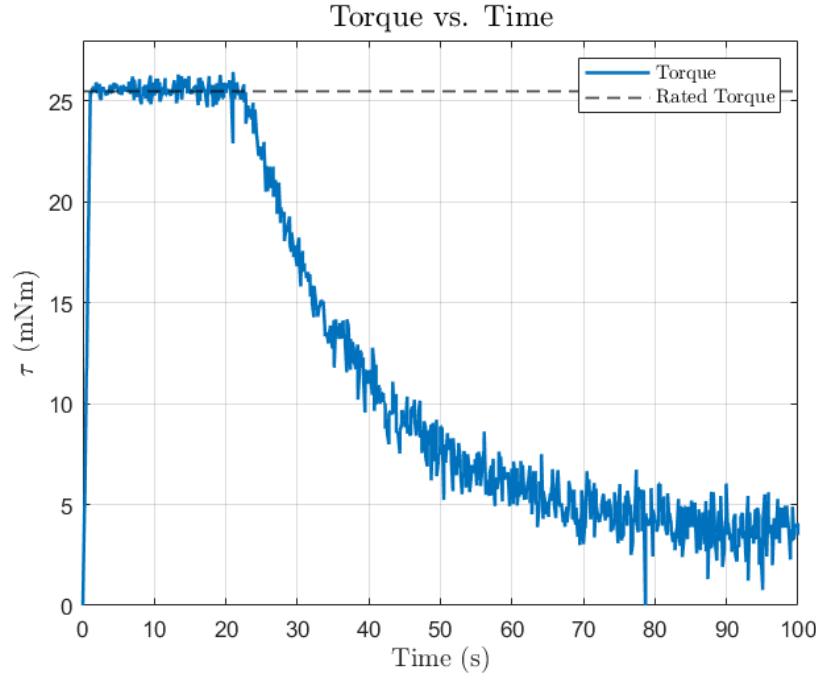
Figure 4.5: SADS Testing Platform

The SADS platform requires balancing in order to emulate the space environment as precisely as possible. A mass balancing system has been the focus of past and current students, and was utilized to balance the system with just a single reaction wheel mounted on it. The balancing was achieved with the help of the current student working on the balance system, Cameron Zorio. Although the system was relatively balanced before conducting the test, there were still some slight errors that propagated

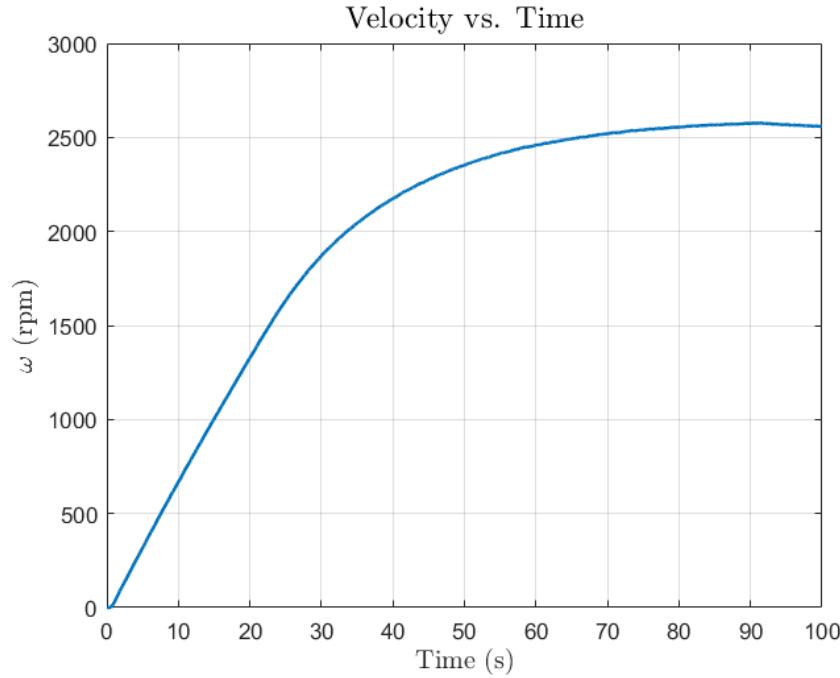
to the IMU data during the test, which will be discussed below. The balancing algorithm was performed after the single reaction wheel was mounted, but before the reaction wheel began accelerating. During the test, when the reaction wheel was performing an actuation, the mass balancing algorithm was turned off.

The maximum wheel speed ω_{wheel} achieved during this test was 2577 rpm or 269.86 rad/s, as seen in fig. 4.6b. Thus, from eq. (2.2), the angular momentum capacity of the wheel is 0.463 N m s, yielding an error of 7.36 % compared to the design value. This obviously represents a significant deviation from design value, and there are many factors that could contribute to this. Firstly, the specified max wheel speed of the motor of 2760 rpm may have been for no load, which may deviate from the actual value with a load attached. Additionally, friction was present in the system through the bearing, which would impact the maximum wheel speed. Moreover, the wheel is located inside a tight-fit enclosed housing, which may lead to air drag disturbances versus an open-air reaction wheel.

The decrease in torque seen in fig. 4.6a was not due to a decrease in input torque, rather the effects of back electromotive force (EMF), a voltage generated in electric motors that opposes the applied voltage, thus decreasing torque input. This performance is expected per motor specifications. To verify that the drop in torque was due to back EMF, the same test was performed on the reaction wheel alone, not mounted to the SADS. The resulting torque outputs for both the SADS test and the reaction wheel alone were plotted in fig. 4.8. As observed, the effects of back EMF are very similar, and the slight differences are likely due to the mass properties associated with having the reaction wheel mounted on the SADS versus by in a stand-alone configuration. Another contributing factor could be that the SADS test was powered with a 24V battery, and the stand-alone test was powered by a power supply.



(a) Torque



(b) Wheel Speed

Figure 4.6: Motor Data for SADS Test

In the IMU data, one can observe that the platform begins at rest, and once a torque is applied through the reaction wheel, motion on the platform begins. The platform

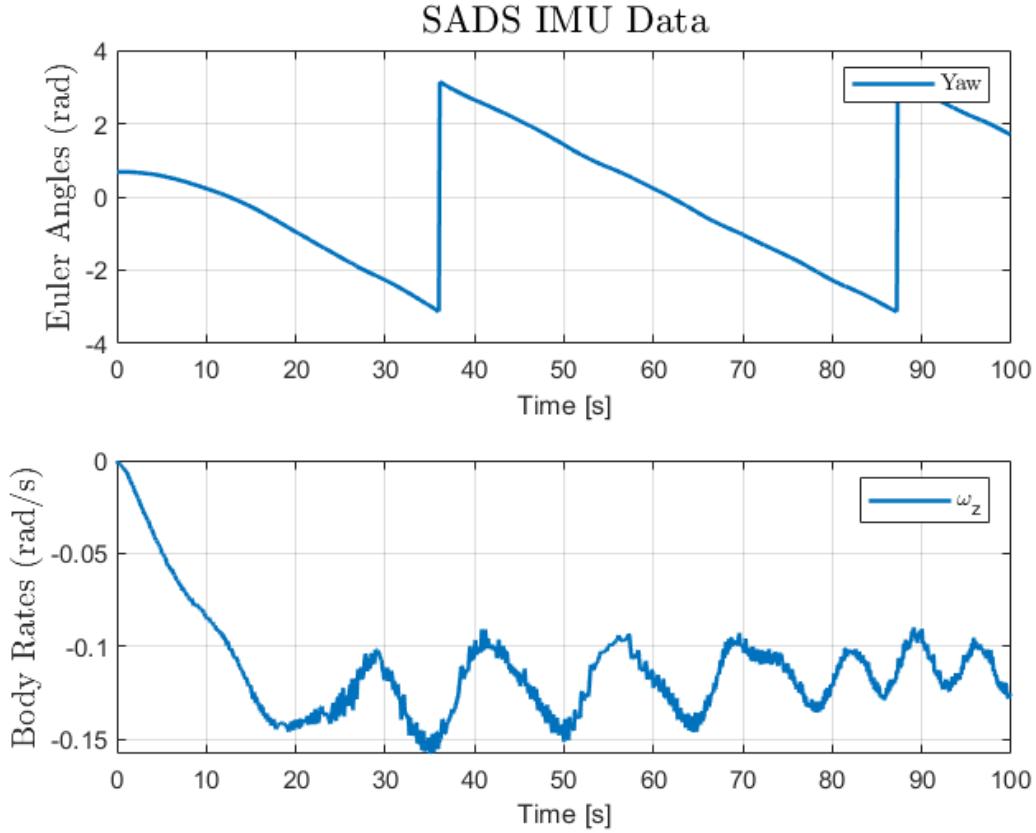


Figure 4.7: IMU Data from SADS Test

begins accelerating until the 20-second mark, where it hits an oscillatory body rate as seen in fig. 4.7. This is likely due to imbalance issues caused by the single-reaction wheel imparting a torque that moves the center of mass. When the SADS center of mass is not perfectly centered on the hemispherical air bearing on which it is mounted, any displacement out of the yaw-plane would move the center of mass off the air bearing, inducing an external torque and effectively resulting in a pendulum-like motion in the system. This would result in an oscillatory behavior as the wheel attempts to further actuate the platform while the external torque attempts to oppose this action. If the SADS were able to rotate freely with three degrees of freedom, it would continue accelerating until the wheel reached saturation, which from fig. 4.6b is approximately 90 seconds. The SADS has limits of travel of approximately $\pm 20^\circ$ out of plane, which restricts the full three degrees of freedom.

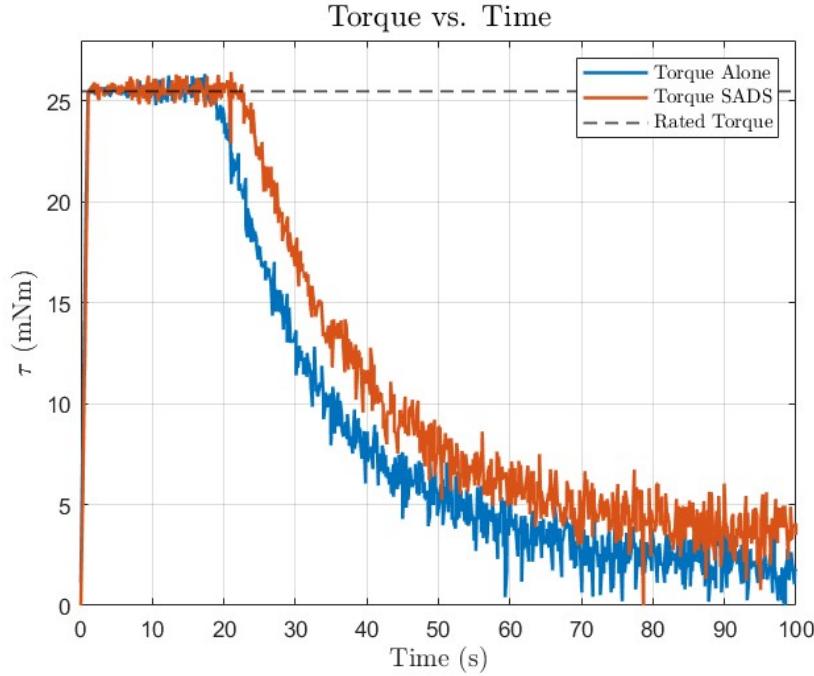


Figure 4.8: Back EMF Comparison

By utilizing a similar linear regression technique to the one described in section 4.2.1, the angular acceleration of SADS in the yaw direction was determined to be 9.04×10^{-3} rad/s². The linear regression only took into account the first 10 seconds of the test, as that data was before oscillation began to occur. From the Earth tracking scenario in section 2.3, the angular acceleration goal derived for the SADS was $\dot{\omega}_{sat} = 7.23 \times 10^{-5}$ rad/s². Clearly, even a single reaction wheel provides sufficient torque to meet the required angular acceleration in the yaw direction.

4.2.3 Torque Profile Test

During initialization of this project, it was desired to perform a torque profile test to demonstrate that a specific position of a system could be controlled via a reaction wheel. This test demonstrates the ability to generate a controlled torque profile, resulting in a controlled position. The test was conducted with the reaction wheel

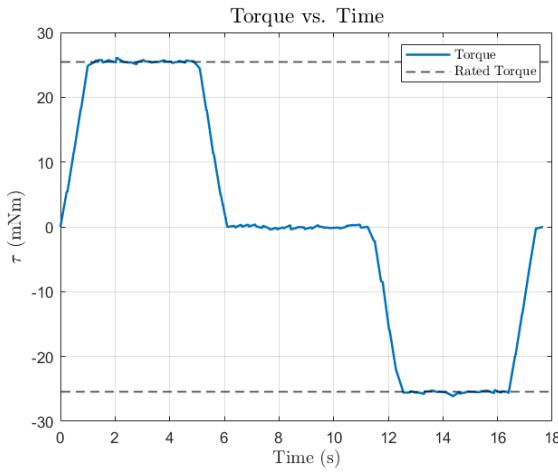
position and velocity, however this would result in similar position and velocity profiles on a system that the reaction wheel was mounted on.

A positive rated torque was applied for 5 seconds, then the wheel was allowed to coast for 5 seconds, then a negative rated torque was applied for 5 seconds as shown in fig. 4.9a. This results in a set position value. On an actual system, this position value could be set beforehand, and the reaction wheel would be capable of reaching this value by performing a torque profile similar to this. There is some error in the graphs of fig. 4.9, especially due to the coast period resulting in a slowing wheel speed as shown in fig. 4.9b (due to friction and other energy losses during that period).

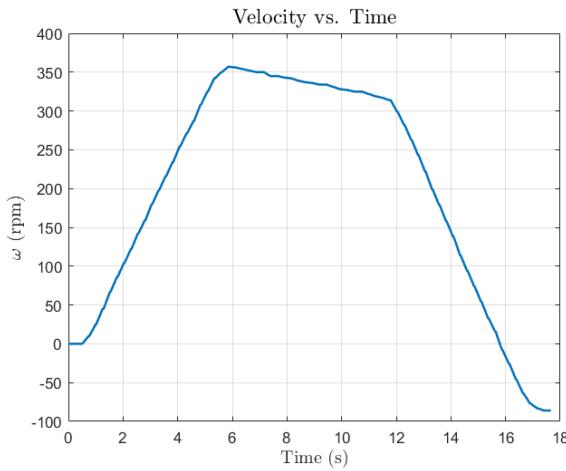
4.2.4 Sources of Error

As previously mentioned, there were sources of error present in this testing configuration that affected output values. As far as instrumentation, the hall sensors were utilized for measurement values of torque and velocity from the motor, and position values were integrated from velocity. The hall sensors provide adequate data for initial testing, however motor encoders would increase the precision of the data. These tests would be improved with higher precision equipment. Also, outside equipment with known values could also be utilized to test values such as inertia, however the tests were conducted with available equipment at the time. Each test was repeated multiple times to ensure accurate values as well.

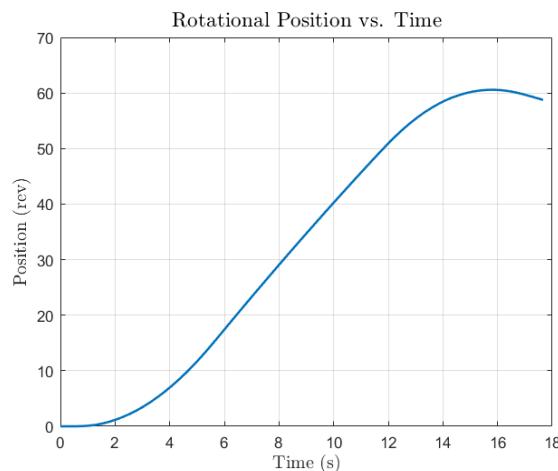
For testing on the SADS, more significant results would be able to be achieved with all four wheels manufactured and working in conjunction. The single wheel test demonstration was helpful for basic capabilities, but the tests of interest for actual spacecraft control would require all four wheels. Additionally, the reaction wheel and mass balancing system were run with two separate controllers. In the future, having



(a) Torque Profile



(b) Angular Velocity Profile



(c) Angular Position Profile

Figure 4.9: Plots for Torque Profile

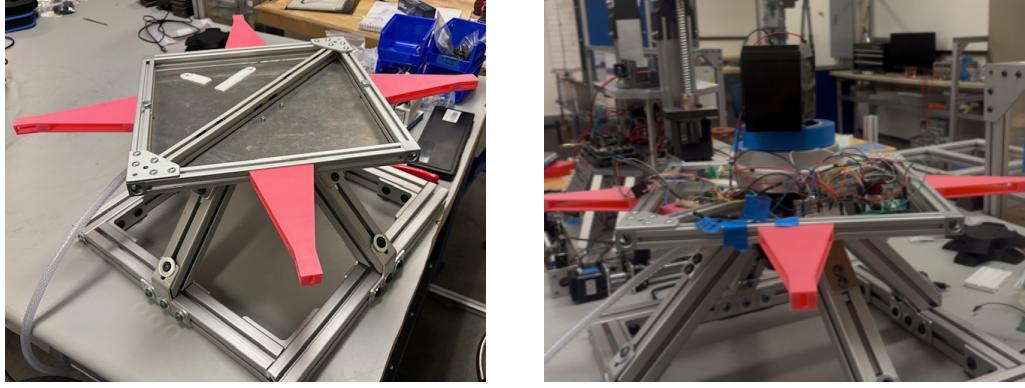
a central flight computer controlling the entire system would allow for communication between subsystems and increased performance.

4.3 Control Demonstration

For additional testing and demonstration, the Spacecraft Attitude Control Simulator (SACS) was utilized to allow the reaction wheel to interface with an outside system. The SACS a single-axis attitude control platform which is intended to recreate the frictionless space environment [21]. This platform was constructed by Bricen Rigby, a fellow student working with Dr. Mehiel. The platform creates a low friction environment by utilizing an air-bearing, maintained at a constant pressure by a regulator and pump set-up. A platform was placed on the air spindle to hold the reaction wheel and associated electronics, as pictured in fig. 4.10a.

The reaction wheel was mounted on a T-slotted aluminum extrusion cross beam via a 3D printed mount, which sufficed for initial testing as shown in fig. 4.10b. For future testing, it would be beneficial to manufacture a higher-strength aluminum mount to reduce potential vibration sources. The electronics were left on the platform for prototyping and rapid adjustments, however for the future use of this wheel, these would be encapsulated within the reaction wheel system. A 24V battery was used to provide power on the platform.

The reaction wheel system was not developed for the SACS platform (it was developed for the SADS), so extensive testing was not performed with this set-up. However, it did provide a method to demonstrate the control capabilities of a single wheel, versus at least three being required with the SADS (and four for full functionality).



(a) SACS Platform

(b) SACS with Reaction Wheel

Figure 4.10: SACS Testing Platform

When a torque command was sent to the wheel, resulting motion was able to be observed on the SACS, demonstrating the capability of the wheel. Performing a constant torque test demonstrated the capabilities of the wheel to accelerate a platform. Of more interest was implementing a simple PID controller to demonstrate basic control capabilities as well. It utilized the IMU data to generate a control torque command, and this command was sent via an analog signal to the motor controller as previously discussed. The Arduino code for the PID control and data collection is included in appendix B.

This test was performed to demonstrate the capabilities of the reaction wheel to control another system. In principle, the reaction wheel was designed for the SADS and the performance of the reaction wheel with the SADS is of more interest. However, this test validates the capabilities of the reaction wheel to perform closed-loop control maneuvers. For similar testing on the SADS, four reaction wheels will have to be manufactured and assembled.

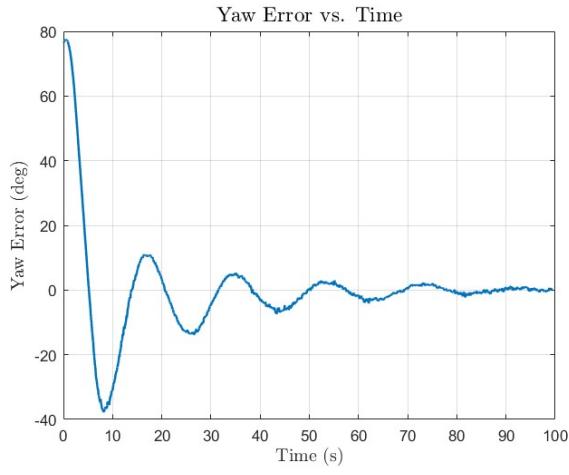
The PID test run on the SACS resulted in the graphs shown in fig. 4.11. For this test, IMU position data of the platform was collected, as well as motor torque and velocity values. This test utilized a laptop on the platform for data collection, significantly

increasing the inertia of the platform and effecting settling times and balance of the platform overall. From fig. 4.11a, it is clear that the system is capable of performing closed-loop control to a specified point even with the added inertia. Due to the imbalance in the system, there is a resulting disturbance torque that must be counteracted, resulting in the nonzero torque seen in fig. 4.11b even after the platform has settled. This results in the increase in motor velocity seen in fig. 4.11c. On a perfectly balanced system, the torque should approach zero as the platform settles, and the wheel would remain at a relatively constant velocity.

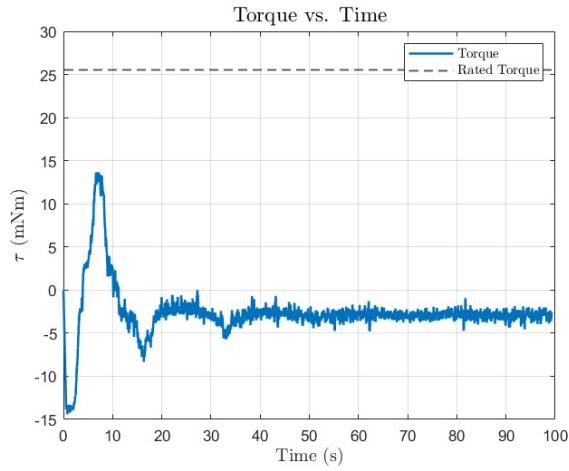
This test was performed to validate the closed-loop capabilities of the reaction wheel system, however was not meant to validate performance metrics. The PID response demonstrates an underdamped controlled response, which could be further tuned with additional work. However, because the reaction wheel was not designed for SACS, fine-tuning performance for this test was not a desired result.

4.4 Software and Testing Conclusion

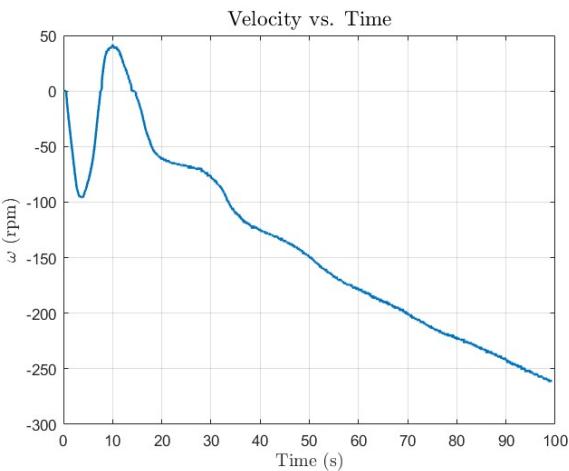
To conclude, the testing specifications will be compared to the original design requirements discussed in chapter 2. The collection of the key requirements are tabulated in table 4.1. The N/A for SADS angular acceleration is present due to the fact that an increased angular acceleration capability was not a design error, rather a result of having a motor capable of producing more torque than the system required. Being able to perform faster angular accelerations than required by the ground tracking scenario posed earlier would allow for more visible tests to be conducted, and giving a larger system capability for future experimentation overall. Additionally, the reaction wheel system demonstrated closed-loop control capabilities with a PID controller. Future



(a) IMU Position Data



(b) Motor Torque Data



(c) Motor Velocity Data

Figure 4.11: Plots for PID Test on SACS

work on this system will require closed-loop capabilities, and verifying functionality will allow for a wider range of control laws to be tested.

Table 4.1: Key Reaction Wheel Tested Specifications

Reaction Wheel Tested Specifications			
Parameter	Design Value	Tested Value	% Error
Momentum Capacity (N m s)	0.500	0.463	7.36
Wheel Inertia (kg m ²)	1.698×10^{-3}	1.716×10^{-3}	1.08
Max Wheel Speed (rpm)	2760	2577	6.63
SADS Angular Acceleration (rad/s ²)	7.23×10^{-5}	9.04×10^{-3}	N/A

Chapter 5

CONCLUSION

The work in this thesis created one fully operational independent reaction wheel system designed for the SADS, validated the wheel specifications, and created a repeatable methodology that can be utilized for creating reaction wheels, meeting the thesis objectives.

There remains significant scope for future work on the reaction wheel system and SADS as a whole. On the reaction wheel, a full-fledged characterization of the wheel could be performed, implementing more testing equipment and rigorous procedures to determine desired wheel characteristics. Important wheel characteristics to test would be static and dynamic imbalance, vibration frequencies and associated wheel speeds, and friction between the assembly and wheel as previously. After these tests were performed, additional revisions to the design could be implemented, further increasing performance of the wheel.

The mechanical design of the wheel and associated assembly parts could also be further improved with further analysis. Stress analysis, vibration analysis, and other mechanical tests could lead to lighter, more optimized components for the assembly. Additionally, the overall geometry of the assembly could be improved as wheel, such as having the motor be flush with the housing instead of protruding as it currently does. Accounting for housing the electronics in the assembly is an important design consideration as well, which could be achieved with further design improvements and efforts into customized electronics such as a PCB for the system. When a desired final reaction wheel design is set, three more will have to be replicated for full usage

of a pyramidal reaction wheel configuration. Once a final design has been reached, a full interface control document (ICD) for the reaction wheel system would provide clear documentation for the replication of the system.

The single reaction wheel in conjunction with the SACS platform provides an educational tool to demonstrate control capabilities, as well as a visual representation of the technology utilized in modern day spacecraft. Additional work could make this system accessible to students to program and use for experiments.

Once four reaction wheels are manufactured and assembled, full control testing could be performed on the SADS, testing the capabilities of the reaction wheels on the system they were designed for. Once operational with the SADS, a multitude of tests could be performed for system actuation of the SADS via the reaction wheel system, demonstrating the capabilities of the system as a whole. An operational SADS would allow for ground testing of control laws, equipment, and dynamics of a spacecraft. This would also function as a great educational tool, allowing for additional experimentation and learning.

Overall, this project serves as a base model for reaction wheel design and demonstrates the system functions as designed. This thesis should be able to be replicated and modified for systems other than the SADS, and future work will be able to reference this project to support development and system design.

BIBLIOGRAPHY

- [1] *5.0 Guidance, Navigation, and Control - NASA*. URL: <https://www.nasa.gov/smallsat-institute/sst-soa/guidance-navigation-and-control/> (visited on 11/12/2024).
- [2] *Brushless DC Flat Motor: DF32A — US Online-Shop*. URL: <https://www.nanotec.com/us/en/products/2139-df32m024027-a> (visited on 03/29/2025).
- [3] D. M. Burkhardt, J. R. Snider, and J. A. Firth. “Flight Qualification for Low Cost Student Manufactured and Balanced Reaction Wheel”. In: *2023 Regional Student Conferences*. AIAA Regional Student Conferences. American Institute of Aeronautics and Astronautics, Jan. 2023. doi: 10.2514/6.2023-70744. URL: <https://arc-aiaa-org.calpoly.idm.oclc.org/doi/10.2514/6.2023-70744> (visited on 11/17/2024).
- [4] A. Calvi. *ECSS-E-HB-32-26A Spacecraft Mechanical Loads Analysis Handbook*. ResearchGate. URL: https://www.researchgate.net/publication/276266312_ECSS-E-HB-32-26A_Spacecraft_Mechanical_Loads_Analysis_Handbook (visited on 11/17/2024).
- [5] *CL3-E-1-0F - Motor Controllers / Drives — Nanotec*. URL: <https://www.nanotec.com/us/en/products/1758-cl3-e-1-0f> (visited on 04/11/2025).
- [6] R. Cowen. “The Wheels Come off Kepler”. In: *Nature* 497.7450 (May 1, 2013), pp. 417–418. ISSN: 1476-4687. doi: 10.1038/497417a. URL: <https://www.nature.com/articles/497417a> (visited on 10/31/2024).
- [7] L. Dam. “Applied Mass Properties Identification Method to the Cal Poly’s Spacecraft Simulator”. In: *Master’s Theses* (Apr. 1, 2014). doi: 10.15368/theses.2014.25. URL: <https://digitalcommons.calpoly.edu/theses/1175>.

- [8] C. Dennehy. “A Survey Of Reaction Wheel Disturbance Modeling Approaches For Spacecraft Line-of-sight Jitter Performance Analysis”. In: 2019. URL: <https://www.semanticscholar.org/paper/A-SURVEY-OF-REACTION-WHEEL-DISTURBANCE-MODELING-FOR-Dennehy/> (visited on 11/17/2024).
- [9] M. Downs. “Adaptive Control Applied to the Cal Poly Spacecraft Attitude Dynamics Simulator”. In: *Master’s Theses* (Feb. 1, 2010). DOI: 10.15368/theses.2010.6. URL: <https://digitalcommons.calpoly.edu/theses/231>.
- [10] *Fundamentals of Spacecraft Attitude Determination and Control — PDF — Attitude Control — Physics*. Scribd. URL: <https://www.scribd.com/document/321156353/Fundamentals-of-Spacecraft-Attitude-Determination-and-Control> (visited on 10/30/2024).
- [11] A. Gilman. “Automatic Mass Balancing Of A Spacecraft Attitude Dynamics Simulator With Six Sliding Masses”. In: *Master’s Theses* (June 1, 2024). URL: <https://digitalcommons.calpoly.edu/theses/2858>.
- [12] M. D. Hasha. “High-Performance Reaction Wheel Optimization for Fine-Pointing Space Platforms: Minimizing Induced Vibration Effects on Jitter Performance plus Lessons Learned from Hubble Space Telescope for Current and Future Spacecraft Applications”. In: NASA Ames Research Center. Moffett Field, CA, United States, May 4, 2016. URL: <https://ntrs.nasa.gov/citations/20160008147> (visited on 11/16/2024).
- [13] T. Kato. “Modification of the Cal Poly Spacecraft Simulator System for Robust Control Law Verification”. In: *Master’s Theses* (June 1, 2014). DOI: 10.15368/theses.2014.48. URL: <https://digitalcommons.calpoly.edu/theses/1201>.
- [14] R. Kinnett. “System Integration and Attitude Control of a Low-Cost Spacecraft Attitude Dynamics Simulator”. In: *Master’s Theses* (Mar. 1, 2010). DOI: 10.15368/theses.2010.40. URL: <https://digitalcommons.calpoly.edu/theses/271>.

- [15] K.-C. Liu, P. Maghami, and C. Blaurock. “Reaction Wheel Disturbance Modeling, Jitter Analysis, and Validation Tests for Solar Dynamics Observatory”. In: *AIAA Guidance, Navigation and Control Conference and Exhibit*. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 18, 2008. doi: 10.2514/6.2008-7232. URL: <https://arc.aiaa.org/doi/10.2514/6.2008-7232> (visited on 11/15/2024).
- [16] F. L. Markley and J. L. Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*. New York, NY: Springer, 2014. doi: 10.1007/978-1-4939-0802-8. URL: <http://link.springer.com/10.1007/978-1-4939-0802-8> (visited on 10/30/2024).
- [17] R. A. Masterson, D. W. Miller, and R. L. Grogan. “Development and Validation of Reaction Wheel Disturbance Models: Empirical Model”. In: *Journal of Sound and Vibration* 249.3 (Jan. 17, 2002), pp. 575–598. ISSN: 0022-460X. doi: 10.1006/jsvi.2001.3868. URL: <https://www.sciencedirect.com/science/article/pii/S0022460X01938684> (visited on 11/15/2024).
- [18] C. Mittelsteadt and E. Mehiel. “Cal Poly Spacecraft Attitude Dynamics Simulator: CP/SADS”. In: *AIAA Guidance, Navigation and Control Conference and Exhibit*. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 20, 2007. doi: 10.2514/6.2007-6443. URL: <https://arc.aiaa.org/doi/10.2514/6.2007-6443> (visited on 10/10/2024).
- [19] *Non-Inverting Op Amp — Analog Devices*. URL: <https://www.analog.com/en/resources/glossary/non-inverting-op-amp.html> (visited on 05/05/2025).
- [20] *Reaction Wheels Components Blue Canyon Technologies*. Blue Canyon Technologies. URL: <https://www.bluecanyontech.com/components/reaction-wheels/> (visited on 04/11/2025).

- [21] B. Rigby. “Spacecraft Attitude Control Simulator”. In: *College of Engineering Summer Undergraduate Research Program* (Oct. 1, 2024). URL: https://digitalcommons.calpoly.edu/ceng_surp/50.
- [22] A. Sabnis, F. Schmitt, and L. Smith. *Magnetic Bearing Reaction Wheel*. NASA-CR-148315. Feb. 1, 1976. URL: <https://ntrs.nasa.gov/citations/19760020248> (visited on 11/12/2024).
- [23] S. Silva. “Applied System Identification for a Four Wheel Reaction Wheel Platform”. In: *Master’s Theses* (June 1, 2010). doi: 10.15368/theses.2010.97. URL: <https://digitalcommons.calpoly.edu/theses/328>.

APPENDICES

Appendix A

LIST OF ACRONYMS

CMG Control-Moment Gyros

CNC Computer Numerical Control

GNC Guidance, Navigation, and Control

ICD Interface Control Document

IMU Inertial Measurement Unit

LEO Low Earth Orbit

MBRW Magnetic Bearing Reaction Wheels

MED Momentum-Exchange Device

PWM Pulse Width Modulation

PWRP Pyramidal Reaction Wheel Platform

SACS Spacecraft Attitude Control Simulator

SADS Spacecraft Attitude and Dynamics Simulator

Appendix B

CODE

Code Listing B.1: NanoJ Torque Control

```
// You can map frequently used objects to be able to
// read or write them
// using In.* and Out.*. Here we map the object 6041:00
// as "In.StatusWord".
map U16 Controlword as inout 0x6040:00
map U16 Statusword as input 0x6041:00
map U32 Inputs as input 0x60FD:00
map U32 Outputs as inout 0x60FE:01
map S08 ModesOfOperation as output 0x6060:00
map S08 ModesOfOperationDisplay as input 0x6061:00
map S16 AnalogInput as input 0x3220:01
map S16 TargetTorque as inout 0x6071:00

// Include the definition of NanoJ functions and
// symbols
#include "wrapper.h"
#include "nanotec.h"

// The user() function is the entry point of the NanoJ
// program. It is called
// by the firmware of the controller when the NanoJ
// program is started.
void user()
{
    ModesOfOperation(4); // set the mode of
    // operation to "profile torque"

    // Enable voltage, set target torque, set max. torque,
    // set torque slope

    Shutdown(); // set the state machine to "
    ReadyToSwitchOn"
```

```

od_write(0x6072, 0x00, 1000); // set maximum torque to
    100% of the rated torque

od_write(0x6087, 0x00, 1000); // set torque slope to
    100% of the rated torque within 1s

// Max Torque and Offset
S32 MaxTorque = 1000;

// Use input 5 as "start"

yield();

while(true)
{
    while(DigitalInput(5))
    {
        // Scale the torque up to the Max Torque in a
        // linear relationship with analog voltage

        if(DigitalInput(1)) // in a HIGH state,
            positive torque
        {
            InOut.TargetTorque = MaxTorque *
                AnalogInput() / 1023;
        }
        else // in a LOW state, negative torque
        {
            InOut.TargetTorque = -MaxTorque *
                AnalogInput() / 1023;
        }

        EnableOperation();
    }

    Shutdown();
    yield();
}
}

```

Code Listing B.2: Arduino CANopen and SD Card Communication

```
#include <SPI.h>
```

```

#include <SD.h>
#include "mcp_can.h"
#include "MPU9250.h"

#define CAN_CS 10
#define SD_CS 7

MPU9250 mpu;

const int pwmPin = 9;           // PWM output: torque
                               magnitude
const int dirPin = 8;          // Digital output: direction
const int switchPin = 4;        // Control motor controller
                               on/off

MCP_CAN CAN(CAN_CS);

const byte NODE_ID = 127;
const unsigned long REQ_ID = 0x600 + NODE_ID;
const unsigned long RESP_ID = 0x580 + NODE_ID;

// Velocity (606Ch:00)
const byte INDEX_LOW = 0x6C;
const byte INDEX_HIGH = 0x60;
const byte SUBINDEX = 0x00;

// Torque (6077h:00)
const byte TORQUE_INDEX_LOW = 0x77;
const byte TORQUE_INDEX_HIGH = 0x60;

// Position (6064h:00)
const byte POS_INDEX_LOW = 0x64;
const byte POS_INDEX_HIGH = 0x60;

unsigned long lastLogTime = 0;
const unsigned long logInterval = 200; // Log every
                                     200 ms

int32_t velocity = 0;
int16_t torque = 0;

enum ReadState { READ_VELOCITY, READ_TORQUE };
ReadState currentState = READ_VELOCITY;

void setup() {

```

```

Serial.begin(115200);
delay(1000);

pinMode(switchPin, OUTPUT);
digitalWrite(switchPin, HIGH);

pinMode(dirPin, OUTPUT);
digitalWrite(dirPin, HIGH);

pinMode(CAN_CS, OUTPUT);
pinMode(SD_CS, OUTPUT);
digitalWrite(CAN_CS, HIGH);
digitalWrite(SD_CS, HIGH);

SPI.begin();

// Init SD
if (!SD.begin(SD_CS)) {
    Serial.println("SD init failed");
    while (1);
}
Serial.println("SD init OK");

Serial.println("Creating log file...");
File dataFile = SD.open("test.csv", FILE_WRITE);
if (dataFile) {
    dataFile.println("TimeMS,Velocity,Torque");
    dataFile.close();
    Serial.println("Log file created.");
} else {
    Serial.println("Failed to create log file");
    while (1);
}

// Init CAN
if (CAN.begin(MCP_ANY, CAN_1000KBPS, MCP_8MHZ) != CAN_OK) {
    Serial.println("CAN init failed");
    while (1);
}
CAN.setMode(MCP_NORMAL);
Serial.println("CAN init OK");

delay(100);

```

```

setupController();
delay(100);

requestVelocity(); // Start the cycle
}

void loop() {
    unsigned char len = 0;
    unsigned char buf[8];
    unsigned long canId = 0;

    if (CAN_MSGAVAIL == CAN.checkReceive()) {
        CAN.readMsgBuf(&canId, &len, buf);

        if (canId == RESP_ID) {
            // Velocity response
            if (buf[0] == 0x43 && buf[1] == INDEX_LOW && buf
                [2] == INDEX_HIGH && buf[3] == SUBINDEX) {
                velocity = (int32_t)((uint32_t)buf[4] |
                                      ((uint32_t)buf[5] << 8) |
                                      ((uint32_t)buf[6] << 16) |
                                      ((uint32_t)buf[7] << 24));
                currentState = READ_TORQUE;
                delay(5);
                requestTorque();
            }

            // Torque response
            else if (buf[0] == 0x4B && buf[1] ==
                     TORQUE_INDEX_LOW && buf[2] ==
                     TORQUE_INDEX_HIGH && buf[3] == 0x00) {
                torque = (int16_t)(buf[4] | (buf[5] << 8));

                unsigned long now = millis();
                Serial.print("Time: "); Serial.print(now);
                Serial.print(" ms | Vel: "); Serial.print(
                    velocity);
                Serial.print(" | Trq: "); Serial.println(torque
                );
            }
        }
    }

    File dataFile = SD.open("test.csv", FILE_WRITE)
    ;
    if (dataFile) {
        dataFile.print(now);
        dataFile.print(",");
    }
}

```

```

        dataFile.print(velocity);
        dataFile.print(",");
        dataFile.println(torque);
        dataFile.close();
    }

    currentState = READ_VELOCITY;
    lastLogTime = millis();
}

// SDO Abort
else if (buf[0] == 0x80) {
    uint16_t index = (buf[2] << 8) | buf[1];
    uint32_t code = ((uint32_t)buf[4]) |
                    ((uint32_t)buf[5] << 8) |
                    ((uint32_t)buf[6] << 16) |
                    ((uint32_t)buf[7] << 24);
    Serial.print("SDO Abort at index 0x");
    Serial.print(index, HEX);
    Serial.print(", code 0x");
    Serial.println(code, HEX);
}
}

// Control request pacing
if (millis() - lastLogTime >= logInterval &&
    currentState == READ_VELOCITY) {
    requestVelocity();
}
}

void requestVelocity() {
    byte request[8] = {
        0x40, INDEX_LOW, INDEX_HIGH, SUBINDEX,
        0x00, 0x00, 0x00, 0x00
    };
    CAN.sendMsgBuf(REQ_ID, 0, 8, request);
}

void requestTorque() {
    byte request[8] = {
        0x40, TORQUE_INDEX_LOW, TORQUE_INDEX_HIGH, 0x00,
        0x00, 0x00, 0x00, 0x00
    };
}

```

```

    CAN.sendMsgBuf(REQ_ID, 0, 8, request);
}

void requestPosition() {
    byte request[8] = {
        0x40, POS_INDEX_LOW, POS_INDEX_HIGH, 0x00,
        0x00, 0x00, 0x00, 0x00
    };
    CAN.sendMsgBuf(REQ_ID, 0, 8, request);
}

void setupController() {
    Serial.println("Configuring drive...");

    sendSDO(0x6060, 0x00, 0x02); // Velocity Mode
    delay(50);
    sendSDO(0x6040, 0x00, 0x0006); // Shutdown
    delay(50);
    sendSDO(0x6040, 0x00, 0x0007); // Switch On
    delay(50);
    sendSDO(0x6040, 0x00, 0x000F); // Enable Operation
    delay(50);
    sendSDO(0x60FF, 0x00, 200); // Target velocity
    delay(50);

    Serial.println("Drive is in Operation Enabled");
}

void sendSDO(uint16_t index, uint8_t subindex, uint32_t
    value) {
    byte len;
    byte data[8];

    if (value <= 0xFF) {
        data[0] = 0x2F; len = 4;
    } else if (value <= 0xFFFF) {
        data[0] = 0x2B; len = 5;
    } else {
        data[0] = 0x23; len = 8;
    }

    data[1] = index & 0xFF;
    data[2] = (index >> 8) & 0xFF;
    data[3] = subindex;
    data[4] = value & 0xFF;
}

```

```

    data[5] = (value >> 8) & 0xFF;
    data[6] = (value >> 16) & 0xFF;
    data[7] = (value >> 24) & 0xFF;

    CAN.sendMsgBuf(REQ_ID, 0, len, data);
}

```

Code Listing B.3: Arduino PID Control

```

#include <SPI.h>
#include "mcp_can.h"
#include "MPU9250.h"

// PIN DEFINITIONS
const int pwmPin      = 9;      // PWM output: torque
                                magnitude
const int dirPin       = 8;      // Digital output: direction
const int switchPin   = 4;      // Motor controller on/off
// const int CAN_CS      = 10;    // SPI chip-select for
                                MCP_CAN
#define CAN_CS 10

// IMU & CAN OBJECTS
MPU9250 mpu;
MCP_CAN CAN(CAN_CS);

// CANOPEN PARAMETERS
const byte NODE_ID     = 127;
const unsigned long REQ_ID  = 0x600 + NODE_ID;
const unsigned long RESP_ID = 0x580 + NODE_ID;
const byte INDEX_LOW   = 0x6C;   // velocity index low
                                byte
const byte INDEX_HIGH  = 0x60;   // velocity index high
                                byte
const byte SUBINDEX    = 0x00;
const byte TORQ_LOW    = 0x77;
const byte TORQ_HIGH   = 0x60;

// PID GAINS & LIMITS
const float Kp = 0.008;
const float Ki = 0.001;
const float Kd = 0.02;
const float maxTorque  = 0.5;    // Nm
const float maxIntegral = 50.0;

```

```

// TIMING
const unsigned long controlInterval = 25;      // ms
const unsigned long logInterval     = 25;      // ms
unsigned long prevControlMs = 0;
unsigned long prevCANReqMs = 0;

// PID STATE
float desiredYaw      = 0.0;
float previousYaw     = 0.0;
float previousTime    = 0.0;
float integralError   = 0.0;

// CAN POLLING STATE
enum ReadState { READ_VELOCITY, READ_TORQUE };
ReadState currentState = READ_VELOCITY;
int32_t velocity = 0;
int16_t torque = 0;

void setup() {
    Serial.begin(115200);
    delay(1000);

    // pins
    pinMode(pwmPin, OUTPUT);
    pinMode(dirPin, OUTPUT);
    pinMode(switchPin, OUTPUT);
    digitalWrite(switchPin, HIGH);

    // IMU init
    Wire.begin();
    delay(2000);
    if (!mpu.setup(0x68)) {
        Serial.println("MPU connection failed.");
        while (1) { delay(500); }
    }
    Serial.println("MPU initialized.");

    // initialize SPI for MCP_CAN
    SPI.begin();

    // CAN init
    if (CAN.begin(MCP_ANY, CAN_1000KBPS, MCP_8MHZ) != CAN_OK) {
        Serial.println("CAN init failed");
        while (1) { }
    }
}

```

```

}

CAN.setMode(MCP_NORMAL);
Serial.println("CAN initialized.");

Serial.println("TimeMS ,Yaw ,Error ,Velocity ,Torque");

delay(100);
setupController();
delay(100);

// kick off first CAN request
prevCANReqMs = millis() - logInterval;
}

void loop() {
    unsigned long now = millis();

    // PID control
    if (mpu.update() && now - prevControlMs >=
        controlInterval) {
        control_loop();
        prevControlMs = now;
    }

    // Handle incoming CAN frames
    if (CAN.checkReceive() == CAN_MSGAVAIL) {
        unsigned char len = 0, buf[8];
        unsigned long canId = 0;
        CAN.readMsgBuf(&canId, &len, buf);

        if (canId == RESP_ID) {
            if (buf[0] == 0x43 && buf[1] == INDEX_LOW && buf
                [2] == INDEX_HIGH) {
                // velocity response
                velocity = (int32_t)((uint32_t)buf[4] |
                    ((uint32_t)buf[5] << 8) |
                    ((uint32_t)buf[6] << 16) |
                    ((uint32_t)buf[7] << 24));
                currentState = READ_TORQUE;
                delay(5);
                requestTorque();
            }
            else if (buf[0] == 0x4B && buf[1] == TORQ_LOW &&
                buf[2] == TORQ_HIGH) {

```

```

        // torque response
        torque = (int16_t)(buf[4] | (buf[5] << 8));
        // print timestamp + vel/trq
        Serial.print(velocity); Serial.print(", ");
        Serial.println(torque);
        currentState = READ_VELOCITY;
    }
    else if (buf[0] == 0x80) {
        // SDO abort
        uint16_t idx = (buf[2] << 8) | buf[1];
        uint32_t code = ((uint32_t)buf[4]) |
                        ((uint32_t)buf[5] << 8) |
                        ((uint32_t)buf[6] << 16) |
                        ((uint32_t)buf[7] << 24);
        Serial.print("SDO Abort idx=0x");
        Serial.print(idx, HEX);
        Serial.print(" code=0x");
        Serial.println(code, HEX);
    }
}
}

// Pace CAN requests @200 ms
if (now - prevCANReqMs >= logInterval && currentState
    == READ_VELOCITY) {
    requestVelocity();
    prevCANReqMs = now;
}
}

// PID control loop
void control_loop() {
    float currentYaw = mpu.getYaw();
    float t = millis() / 1000.0;
    float dt = t - previousTime;
    if (dt <= 0) return;

    // error with wrap
    float error = desiredYaw - currentYaw;
    if (error > 180) error -= 360;
    if (error < -180) error += 360;

    // integral
    integralError += error * dt;
}

```

```

integralError = constrain(integralError, -maxIntegral
, maxIntegral);

// derivative (yaw rate)
float yawRate = (currentYaw - previousYaw) / dt;
if (yawRate > 180.0/dt) yawRate -= 360.0/dt;
if (yawRate < -180.0/dt) yawRate += 360.0/dt;

// PID
float torqueCmd = Kp*error + Ki*integralError - Kd*
yawRate;
torqueCmd = constrain(torqueCmd, -maxTorque,
maxTorque);

// direction & PWM
bool dir = (torqueCmd >= 0);
digitalWrite(dirPin, dir ? HIGH : LOW);
int pwmVal = int(fabs(torqueCmd / maxTorque) * 255);
pwmVal = constrain(pwmVal, 0, 255);
analogWrite(pwmPin, pwmVal);

// print IMU data
Serial.print(millis()); Serial.print(",");
Serial.print(currentYaw, 2); Serial.print(",");
Serial.print(error, 2); Serial.print(",");

// update state
previousYaw = currentYaw;
previousTime = t;
}

// CANopen requests
void requestVelocity() {
byte req[8] = { 0x40, INDEX_LOW, INDEX_HIGH, SUBINDEX
, 0,0,0,0 };
CAN.sendMsgBuf(REQ_ID, 0, 8, req);
}
void requestTorque() {
byte req[8] = { 0x40, TORQ_LOW, TORQ_HIGH, 0x00,
0,0,0,0 };
CAN.sendMsgBuf(REQ_ID, 0, 8, req);
}

// Drive setup sequence
void setupController() {

```

```

Serial.println("Configuring drive...");  

sendSDO(0x6060,0x00,0x02); // Velocity mode  

delay(50);  

sendSDO(0x6040,0x00,0x0006); // Shutdown  

delay(50);  

sendSDO(0x6040,0x00,0x0007); // Switch on  

delay(50);  

sendSDO(0x6040,0x00,0x000F); // Enable operation  

delay(50);  

sendSDO(0x60FF,0x00,200); // Target velocity (example)  

delay(50);  

Serial.println("Drive operational.");  

}  
  

void sendSDO(uint16_t idx, uint8_t sub, uint32_t val) {  

byte len, data[8];  

if (val <= 0xFF) { data[0]=0x2F; len=4; }  

else if (val <= 0xFFFF) { data[0]=0x2B; len=5; }  

else { data[0]=0x23; len=8; }  

data[1]= idx &0xFF;  

data[2]= idx>>8;  

data[3]= sub;  

data[4]= val &0xFF;  

data[5]= (val>>8)&0xFF;  

data[6]= (val>>16)&0xFF;  

data[7]= (val>>24)&0xFF;  

CAN.sendMsgBuf(REQ_ID, 0, len, data);  

}

```
