Design and Analysis for Control Moment Gyroscopes Actuator (CMG) and Testbed for Attitude Control

A project presented to
The Faculty of the Department of Aerospace Engineering
San José State University

In partial fulfillment of the requirements for the degree
Master of Science in Aerospace Engineering

by
Jason Lap Minh Nguyen

May 2025

Approved by

Dr. Yawo Ezunkpe
Faculty Advisor

ABSTRACT

Design and Analysis for Control Moment Gyroscopes Actuator (CMG) and Testbed for Attitude Control

Jason Lap Minh Nguyen

A control moment gyroscope is a device used to orient the attitude of a spacecraft by taking advantage of the conservation of angular momentum through the tilting of a rotating flywheel. This paper documents the process of developing a double-gimbal control moment gyroscope system that enables users to test control algorithms. This development dives deep in deriving the dynamics of the double-gimbal control moment gyroscope by modeling both a version for spacecraft attitude and a proposed testbed. A linear-parameter varying model was used to replicate the nonlinearities that the system exhibits, and was analyzed to determine the stability of the two systems. A gain-scheduling LQR controller was built around this model and would be integrated as the testbed control algorithm. The second half focuses on developing the hardware for the testbed system, where it goes over the hardware and how it was incorporated to work within MATLAB Simulink. Computer-aided design and assembly were provided on the testbed, and approximations were allowed at the moment of inertia. The last stage focused on testing the testbed to validate and verify the dynamic behavior and accuracy of the model.

Acknowledgements

Much acknowledgment goes to those who have provided their assistance with this project. Their dedication and kindness provided unswerving support, fueling a fire of determination that uplifted the goals for this venture. Khyati Mahajan and An Ngo have dedicated much of their time to assisting in CADing and assembling the testbed. Working alongside them was truly a pleasure as we navigated against challenges and sought creative solutions.

I am extremely grateful to Professor Yawo Ezunkpe for being such an inspiring professor in the department. His passion for aerospace is unlike any other, and the amount of support he provided to this project as an advisor helped provide guidance and funding for this project to be possible.

I'd like to express my appreciation to the aerospace department professors, Professor Long Lu and Professor Jeanine Hunter, who inspired my pursuit of controls and dynamics. Their courses and knowledge on these topics laid the foundation for this project and greatly influenced the path of my academic journey in aerospace.

Thank you to the friends I've made along the way. I am immensely grateful for their support throughout my journey at San Jose State University. To my amazing friends - Andreana Aquino, Jesse Franklin, Grace Feng, Derek Ye, Devin Jordao, Stanley Krzeniak, Javaneh Keikha, Hieu Trinh, Joseph Bruno, Richie Lam, and Phuc Phan. With them, they kept my motivation soaring even beyond the stars.

"We evolve beyond the person that we were a minute before.
Little by little, we advance a bit further with each turn.
That's how a drill works!"
- Simon (Tengen Toppa Gurren Lagann)

Table of Contents

List of Figures

List of Tables

Nomenclature

Symbol (s)	Definition	Unit [SI]
n	Orbital Angular Rate	m/s
u_e	Standard Gravitational Parameter of Earth	m^3/s^2
r	Radius from Center Earth to Spacecraft	km
I_X	Moment of Inertia (X-axis)	$kg \cdot m^2$
I_Y	Moment of Inertia (Y-axis)	$kg \cdot m^2$
I_Z	Moment of Inertia (Z-axis)	$kg \cdot m^2$
$R()$	Cosine Rotational Matrix	Unitless
$T_{ext} = M_o$	External Torque	N
H_o	Spacecraft Angular Momentum	$kg \cdot m^2/s$
A	State Matrix	rad/s^2
B	Input Matrix	rad/s^2
C	Output Matrix	rad/s^2
D	Direct Transition	rad/s^2
$x(t)$	State Vector	$rad, rad/s$
$u(t)$	Input Vector	$rad, rad/s$
$\widehat{n}$	Spacecraft-Frame (CMG)	unitless
$\widehat{a}$	Inner Gimbal Frame	unitless
$\widehat{b}$	Flywheel Frame	unitless
h	CMG Angular Momentum	$kg \cdot m^2/s$
$G()$	Jacobian Matrix	unitless
ω	Angular Velocity	rad/s
$\dot{\omega}$	Angular Acceleration	rad/s^2
a	Altitude	km
R	Earth Radius	km
P_n	Pole	unitless
ϕ	Roll	Deg
θ	Pitch	Deg
ψ	Yaw	Deg
$\dot{\phi}$	Roll Rate	Deg/s
$\dot{\theta}$	Pitch Rate	Deg/s
$\dot{\psi}$	Yaw Rate	Deg/s
γ_1	Outer Gimbal Angle	Deg

Symbol (s)	Definition	Unit [SI]
γ_2	Inner Gimbal Angle	Deg
$\dot{\gamma_1}$	Outer Gimbal Angle Rate	Deg/s
$\dot{\gamma_2}$	Inner Gimbal Angle Rate	Deg/s
ρ	LPV Parameter	rad
K	Gain Matrix	unitless
Q_w	State Weighted Matrix	unitless
R_w	Input-cost Weighted Matrix	unitless
R_w	Radius of Flywheel	m
M_W	Mass of Flywheel	kg
α	Filter Coefficient	unitless
a_x	Accelerometer Data (X-axis)	m/s^2
a_y	Accelerometer Data (Y-axis)	m/s^2
a_z	Accelerometer Data (Z-axis)	m/s^2
A_{acc}	Combined Scale Factor Matrix for Accelerometer	unitless
A_{mag}	Combined Scale Factor Matrix for Magnetometer	unitless
B_{acc}	Combined Bias Matrix for Accelerometer	unitless
B_{mag}	Combined Bias Matrix for Magnetometer	unitless
CMG	Control Moment Gyroscope	
SGCMG	Single Gimbal Control Moment Gyroscope	
DGCMG	Double Gimbal Control Moment Gyroscope	
VSCMG	Variable Speed Control Moment Gyroscope	
LVLH	Local Vertical Local Horizontal	
EOM	Equation of Motion	
PWM	Pulse-Width Modulation	
LPV	Linear Parameter Varying	
LTI	Linear Time Invariant	
IMU	Inertial Measuring Unit	
ESC	Electronic Speed Controller	
LIPO	Lithium-ion Polymer	
RCS	Reaction Control Systems	
PLA	Polylactic acid	

1. Introduction

1.1 Motivation

Maintaining stability and orientation is essential when operating a spacecraft during its mission cycle. Aligning the attitude of a spacecraft plays a vital role in various critical subsystems by satisfying system requirements such that solar arrays, observation instruments, and communication antennae operate effectively within their intended environment. This responsibility is operated under the spacecrafts attitude determination and control system (ADCS). A typical ADCS design can actuate and control the current orientation, detumbling the spin of the vehicle if perturbed with some force, and correcting the desired angles over a steady state. Achieving this requires precise maneuvers using actuators that apply corrective torques to correct attitude errors and sensors that can measure yaw, pitch, and roll. Various actuators can be implemented to apply torque, ranging from reaction wheels, reaction control thrusters, solar radiation pressure control panels, and nutation dampers. An actuator known as a control moment gyroscope (CMG) utilizes momentum from a spinning flywheel to correct these attitude angles. By pivoting the flywheel, the momentum axis changes, causing it to produce a torque perpendicular to both the momentum and rotating axes. Integrating CMG can open up various satellite maneuverability options through this method, such as pointing control for Earth observations, local vertical-local horizontal alignment, and target tracking.

Control moment gyroscopes (CMG) were introduced in 1973 on Skylab [16]. Their popularity was more well-known on the ISS, which utilized multiple double-gimbal gyroscopes to achieve attitude control. There are various CMG configurations, each with its benefits and drawbacks. By developing a test bench that can be configured to simulate different varieties of CMGs, the design would help compare various performance characteristics and limitations of the system. A test bench would also provide excellent research opportunities and guidance to the aerospace department at San Jose State University. The majority of aerospace students within academia have primarily only touched on the mathematical models of control theory and thus need more experience in applying the subject to real-world applications. Access to a device would provide students with hardware that would challenge their knowledge and allow them to test the robustness and performance of their controllers. This will enable them to verify their mathematical models and jump-start their innovative designs using the knowledge obtained from this project.

Furthermore, as the popularity of CubeSats has increased during the past decade among institutions and companies, the applications of CubeSats have been sought after by commercial and scientific industries. With the miniaturization of spacecraft, these nanosatellites became commercially affordable and have expanded to applications such as Earth observation [17]. Few CubeSats are designed with actuators and CMGs to achieve high-performance maneuverability for specific applications. For example, satellites capable of imaging are intended to have a slew rate over 0.5 deg/s to track the desired location on Earth and increase the possible imagery obtainable [18]. Recent research on CubeSat focused on single-gimbal CMG; however, the implementation of double-gimbal CMG is still in its infancy. Exploration for a double-gimbal CMG on CubeSats offers more maneuverability than a single-gimbal CMG at the cost of adding complexity to the system.

1.2 Literature Review

1.2.1 Control Moment Gyroscope Fundamentals

Much research has been conducted on the mechanical design of a control moment gyroscope (CMG) over the past decade; however, the primary focus of each literature investigated different forms of CMGs. Nevertheless, CMGs still operate under the same fundamental principles of conservation of angular momentum, where the spacecraft will maintain a constant momentum unless an external force is applied [2]. Angular momentum is stored on a spinning flywheel using a motor to supply the torque and maintain the angular speed. The flywheel is attached to a gimbaled structure that allows perpendicular pitch of the flywheels spin axis. Pitching the flywheel induces a gyroscopic momentum perpendicular to the spin and gimbal axis following the right-hand rule [19]. The magnitude of the gyroscopic momentum is directly proportional to the angular momentum stored in the flywheel and the gimbal rate [2]. With this principle, CMG has been described as a far more efficient method of producing torque than a reaction wheel [3]. Therefore, CMGs offer an efficient torque-to-power ratio, making them suitable ACDS devices for rotational orientation [2].

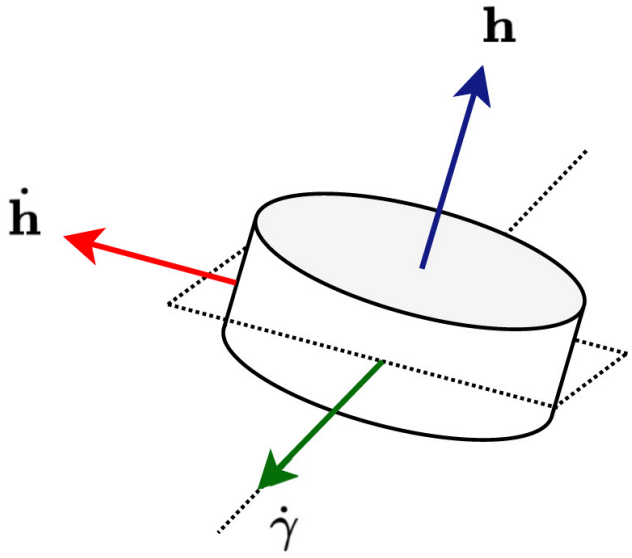

Figure 1.1: CMG illustration on the right-hand rule for the torque output [1]

1.2.1.1 Single Gimbal Control Moment Gyroscope (SGCMG)

Single-gimbal control moment gyroscopes (SGCMGs) are generally known to be the simplest CMGs, as the mechanism and control logic are more accessible to implement. In its basic form, an SGCMG has a single gimbal attached to the flywheel, producing a perpendicular torque that is directional to the cross product of the spin and gimbal axis. The torque is about the frame of the flywheel [20]. The output torque thus can sweep a perpendicular direction in 360 degrees about the gimbal axis [21]. SGCMG is the most power-efficient torque and provides the most torque amplification compared to the other forms of the CMG, but is also limited to the direction in which the torque is produced about the spacecraft [20].

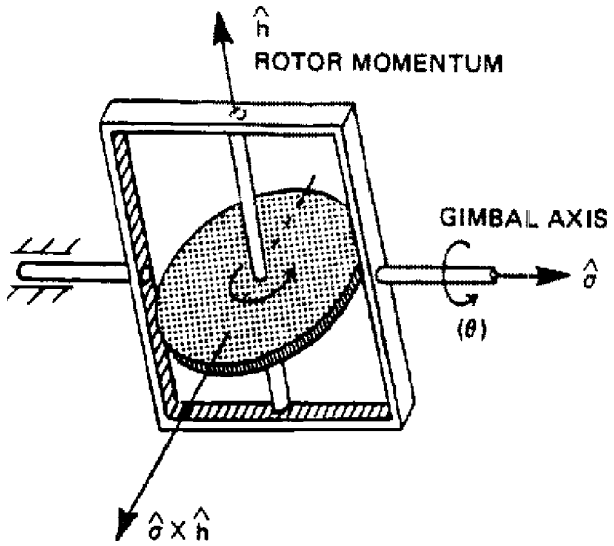

Figure 1.2: Illustration for a SGCMG [2]

1.2.1.2 Dual-Gimbal Control Moment Gyroscope (DGCMG)

Dual-Gimbal Control Moment Gyroscopes are an extension of an SGCMG. DGCMG contains two gimbals, which are referred to as inner and outer gimbals [20]. Each gimbal is mounted on two different axes of the flywheel, driven by two servos. This additional gimbal adds an extra axis to the out torque, giving the DGCMG a two-degree of freedom about the body frame of the flywheel [20, 22]. This provides an advantage over the SGCMG by reducing a phenomenon called singularity on clustered CMG designs, later discussed in 1.2.1 [23]. Thus, DGCMG benefits from better singularity avoidance characteristics than SGCMG but at the cost of the complexity of hardware and control algorithms [24]. Furthermore, references have mentioned that the drawback to the DGCMG is torque amplification, thus requiring more power to operate than an SGCMG [20, 25].

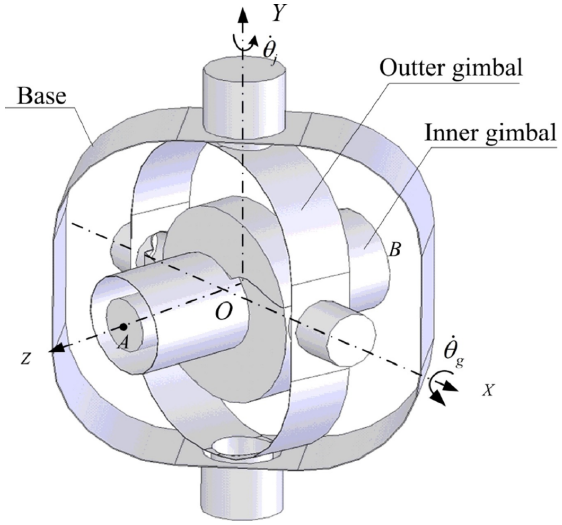

Figure 1.3: Dual-gimbal control moment gyroscope illustration [3]

1.2.1.3 Variable-Speed Control Moment Gyroscope (VSCMG)

VSCMG can be a form of SGCMG or DGCMG designed to change the speed of the flywheel. Typical CMGs spin their flywheels constantly to maintain their angular momentum throughout their operation [26]. One such characteristic of the VSCMG is changing the speed of the flywheel by reducing or increasing the actuation strength of the device [26]. An advantage of the VSCMG is the ability to gain additional freedom within the flywheels body frame. As the flywheel accelerates or decelerates, a control torque is generated parallel to the angular momentum of the flywheel [27]. This is done through a motor and essentially causes the device to inherit characteristics of a reaction wheel. Referencing the literature, VSCMG has been noted to potentially provide more efficiency over controlling attitudes than typical clustered SCMGs, as they require fewer CMGs to operate [27]. Additionally, VSCMG offers a reduction in singularity complexity and disturbances that might influence the momentum within the flywheel [28]. VSCMG comes with challenges as it increases the complexity of designing the controller and requires multiple control algorithms for the flywheel and gimbal to operate [26].

1.2.1.4 CMG Cluster Configuration

A cluster format uses multiple CMGs placed into a specific configuration. Cluster configuration might consist of using SGCMG, DGCMG, or VSCMG. The primary purpose of cluster CMG is to provide additional control over a previously uncontrolled axis. Many types of shapes have been studied, including one-axis scissor control and pyramid cluster. One major drawback caused by using a clustered configuration is the singularity problem complexity that prevents torque from being produced at certain gimbal angles of the CMGs [3, 29]. Adding more CMG can typically help reduce the occurrence of singularities; however, the phenomenon is still persistent, and more CMG would add to the overall mass and size of the system [29]. As stated by the literature, to neglect the issue of singularity, a minimum of six SGCMGs must be used for the spacecraft [28].

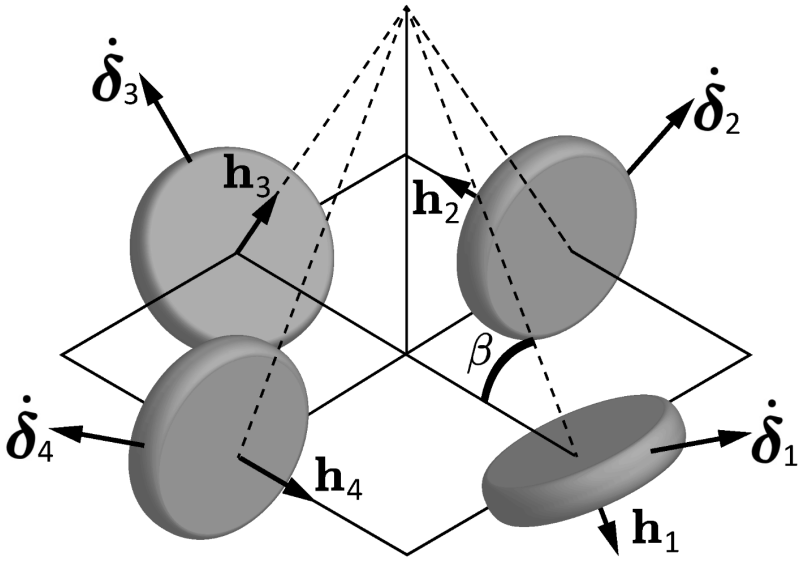

Figure 1.4: Cluster of single gimbal control moment gyroscope illustration [4]

1.2.2 Singularity

When designing a CMG, it is necessary to understand the limitations of how the device can operate. For several reasons, CMGs can encounter a phenomenon in which torque is no longer produced about a specific axis. If the angular momentum of the flywheels is aligned with the axis of rotation, a singularity will happen as the resulting output torque will be perpendicular to the rotation [30]. This can be referred to as a saturation singularity and is classified as the maximum angular momentum about that axis [31]. CMG configurations cause another type of singularity. A cluster of CMGs can be aligned at specific gimbal angles, thus resulting in cancellation in momentum and becoming rank-deficient in its Jacobian Matrix at the point [29]. This is an internal singularity and is generally harder to troubleshoot [27]. There are also external singularities caused by null motion, resulting in it being elliptic [27, 29]. Points indicate internal singularities where total angular momentum is present within the momentum envelope [27].

Engineers use a momentum envelope to analyze the singularities of the system. Though many sources use this momentum envelope, not much information can be obtained that clearly describes the fundamentals behind the concept. From what was gathered, the momentum envelope can distinguish angular momentum feasible directions of the CMG to operate safely within [32]. It is presented as a 3D surface mapping of the possible total angular momentum for the CMG along all rotational directions [32].

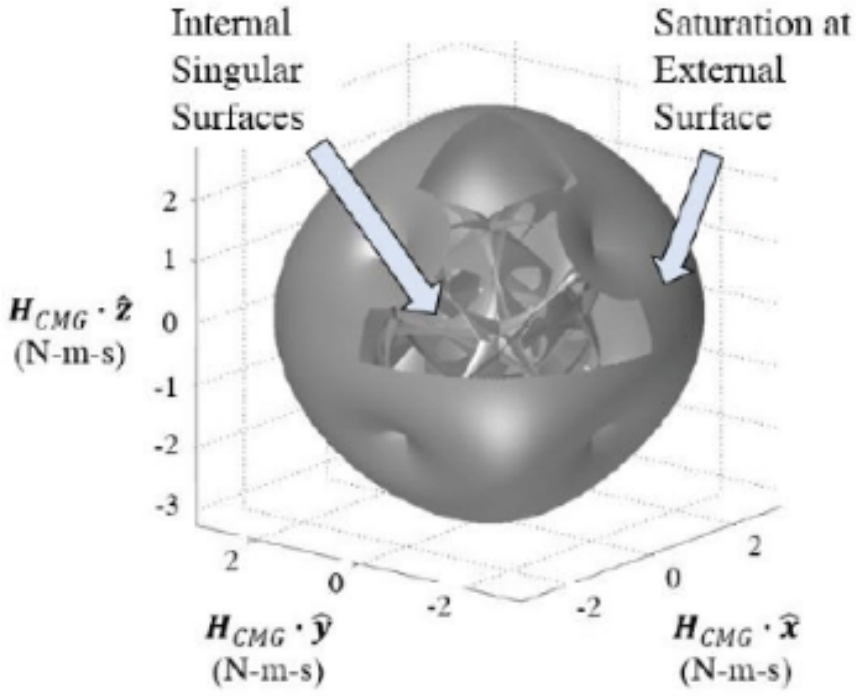

Figure 1.5: Momentum envelope and the external and internal singular surfaces [5]

1.2.3 Steering Law

A steering law is an algorithm introduced to avoid singularities by considering the gimbal angles of the CMGs. Steering control law calculates the specific gimbal rate such that the necessary torque is applied to the spacecraft [28, 30]. The algorithm can apply boundary constraints on the gimbal to model the mechanical constraints that might also be present [33]. Much research has gone into implementing steering laws into various forms of CMGs; thus, many steering control laws have also been studied. Progressive methods, singular direction avoidance, pseudo-inverse logic, and null motion have been used to steer the CMG [34].

Steering laws can be categorized into local and global methods. Local steering laws, such as pseudo-inverse and null-motion algorithms, account for only the current state of the system and are typically used for short-time control [31]. Global Steering Laws are optimized algorithms that try to predict the next state of the spacecraft. Global Steering Laws are generally considered more robust than their counterparts [31].

1.2.4 Controls Systems

When deriving the dynamics of a spacecraft and CMG, the resultant equation of motion will be described as a nonlinear differential equation in 2nd order [35]. The CMG dynamics will also be highly coupled in nature as the torque output is dependent on the gimbal angles [36]. This nonlinearity makes it difficult to apply classical control, thus making it more challenging to design. Approaches in linearizing the equations of motion can be made about an equilibrium point, allowing a linearized model to analyze the stability and locally accurate representation of the nonlinear model to work within a given range. A linear model can tell about the stability of our model at a specific point, but it cannot accurately represent the whole system as a nonlinear model could. This brings up some complications when designing a controller for CMG.

Applying nonlinear controllers can overcome this challenge by removing the need for localization about an equilibrium point from linearization. Recent studies have used nonlinear controllers to achieve stability and the desired outcomes of CMGs. Lyapunov function-based and passivity-based types of nonlinear controllers have been proven to control double-gimbal variable-speed control moment gyroscopes (DGVSCMGs) [25]. A variation of the Lyapunov-based controller for unactuated systems has been tested by Polytechnic School of University researchers Toriumi and Bruno, who analyzed the performance, tracking, and stability of a single-input single-output (SISO) CMG testbench. Establishments in their results have shown feasibility in tracking the CMG and allowed for asymptotic stability [37].

Introducing linear parameter-varying (LPV) has been another alternative to tackling the control problem presented by linearizing the equation of motion. Instead of fully applying linearization methods for the equation of motion, a set of parameters around a set of boundaries can be used to scope multiple linear equations, where each will be locally accurate about a specific condition [38, 39]. A system model can vary its linear equations accordingly based on the operating parameters provided. The result of the system would portray behaviors similar to those of the fully nonlinear model. Because LPV uses a system of scheduled linear equations, linear controllers can be implemented to help obtain stability, but must be tuned appropriately for each linear model. Gaining-scheduling is a well-known and effective technique that accesses tuned gain values from

a linear controller to the corresponding linear model [40]. Tune values are scheduled and called upon when the linear model of the plant changes, ensuring satisfaction over the range of linear models provided by the LPV. Robust controllers like H-infinite synthesis, H2 synthesis, and induced L2-gain have been used to control an LPV system and compute the scheduling gains [41].

1.2.5 Spacecraft Attitude Dynamics

The importance of spacecraft attitude dynamics is primarily used to determine the orientation of the spacecraft in space relative to the vehicle and reference frame. Using the spacecraft body axes motion can allow for an approximation of the respective roll, pitch, and yaw angles [42]. Typically, these angles are aligned with the local orbital reference frame of the spacecraft, where the x-axis is aligned with the velocity vector, and the z-axis is pointed parallel to the nadir [42]. The angles are assigned occur only: yaw is referenced with the z-axis, the roll is referenced with the x-axis, and the pitch is referenced negatively normal to the orbital plane [42]. This configuration can be referred to as local-vertical local-horizontal [42].

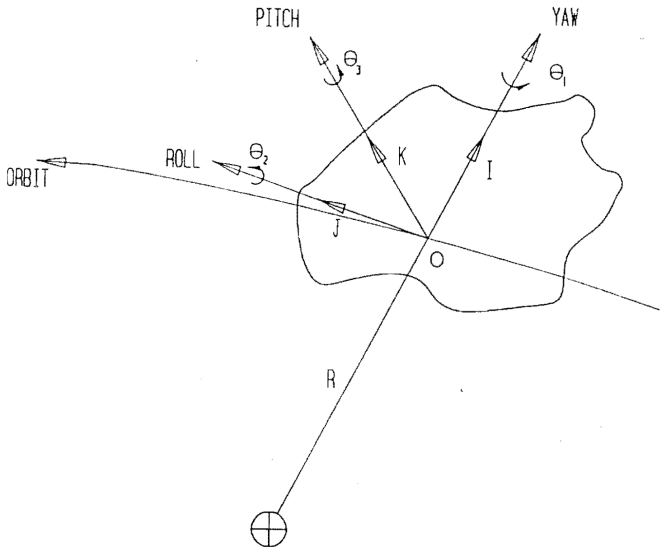

Figure 1.6: Local-vertical local-horizontal attitude coordinate system [6]

Deriving the dynamics of the spacecraft can be referenced in Dynamics of Small Satellites with Gravity Gradient Attitude Control by Julio Zorita. Zorita considered deriving attitude models for low-earth orbits, considering disturbance torques caused by gravity gradient, Earth magnetic, and aerodynamic forces as external forces influencing the system [38]. The spacecraft was treated as a rigid body and thus had a stiff structure throughout. Structures such as booms and solar panels have inherited non-rigid characteristics, but they were assumed to be rigid within this model for simplicity [38]. Applying derivative angular momentum and the sum of external forces, the Euler equation for rigid bodies can be obtained through a few steps demonstrated within Zoritas work [38].

Considering a low-earth orbit (LEO), the most influential disturbance torques present in the environment will be gravitational gradient, aerodynamic forces, and magnetic field. Though the effects from these disturbances are minor, they can induce enough torque to cause the spacecraft

to lose performance or stability in attitude control. Low-earth orbits under 400km altitude will experience more significant aerodynamic torque, which spacecraft dynamics must consider [8]. Neglect is allowed for magnetic disturbances if the magnetic ambient field is significantly small or the spacecraft was initially designed to have few magnetic attractions [8].

One primary interest will be examined is the gravity gradient torque resulting from the distributed mass and gravitational attraction of the spacecraft from the celestial body it orbits. It is an inherently weak torque that engineers need to consider the moment of inertia and attitude of the spacecraft and the celestial body it orbits. Proper design allows a gravity gradient to be used passively to achieve stability at an equilibrium point. In multiple sources, assumptions on gravity gradient must be made to further simplify the model: the satellite is considered to be a rigid body, uniform mass about the satellite, principal axes of the spacecraft are aligned with the reference frame, and the spacecraft is in a circular orbit [8].

1.2.6 Control Moment Gyroscope Testbed

A CMG testbed can be referenced in Experimental Verification for Attitude Control of Spacecraft Utilizing Double-Gimbal Variable-Speed Control Moment Gyroscopes, published by Sayaka Kanata, Futa Hayashida, and Takashi Shimomura. In their work, a Double-gimbal variable-speed control moment gyroscope was designed to allow for three-axis control using two gimbals, each attached to stepper motors to actuate the device [43]. The attitude angles were measured using an IMU attached to the testbed mount [43]. The device was mounted and balanced on a spherical air bearing to reduce friction contact [43]. Their methods indicate that a linear quadratic regulator (LQR) was used to control the roll, pitch, and yaw angles and assumed to have a linear model. A successful test indicated matching results with simulation and experiment, where the CMG device could track the desired attitude angles [43].

1.3 Project Proposal

This project aims to design a control moment gyroscope testbed and allow for analysis of its robustness and performance. Developing a CMG testbench setup is possible using commercially available parts. The motor, electric speed controller, servos, and microcontroller are just a few of the hardware components used to assemble the CMG actuator. Additionally, parts provided by the department, such as the air bearing, can be used to construct the base of the testbed, allowing for a near-frictionless surface during testing. The CMG will focus on a design that can be altered between different variants of CMGs: SGCMG, DGCMG, VSCMG, or a combination. Primarily, a control algorithm will be integrated with the device using specific control methods in which the data can provide information about the performance, tracking, and robustness of the system.

Furthermore, throughout the development of the project, mathematical models will represent satellite attitude control using CMGs at LEO conditions. Another model will be based on the testbed design, which will be compared with the hardware design.

1.4 Methodology

Even though our goal is to design a CMG system, other areas must be addressed to confirm its feasibility. These goals can be methodized below:

1.4.0.1 Derive the Dynamical Model of the CMG and a Simulation

The initial steps in the project will focus on developing two mathematical models that represent a low-earth orbit satellite and a CMG testbed. Models can be derived using rigid body dynamics, knowledge obtained about CMGs, and attitude dynamics of satellites. The equations of motion are expected to be a nonlinear coupled system; thus, possible ways of linearizing the model will be investigated, such as linear parameter varying models (LPV).

Simulations can be achieved by using MATLAB to test the nonlinear model and linearized model of the system. Simulating these models will provide important information about whether the linearized model is an acceptable and feasible method for representing both systems.

1.4.0.2 Implementing Control Methods and Analysis

After designing a controller, designing a control method is critical in driving our linearized model and hardware. The controllability will be analyzed to determine if it can be controlled for all states. Once confirmed, a closed-loop feedback control method will be designed using one of MATLAB extensions, Simulink, where a possible linearized controller, such as PID or LQR, can be applied. Analysis looks at the performance, tracking, and robustness of the controller.

1.4.0.3 Designing Prototype of CMG and Hardware

The structure will be designed using computer-aided design (CAD) software, primarily SolidWorks. Parts of the prototype will be 3D printed with PLA through an FDM printer. The initial design of the CMG will be based on a proposed design suggested by a previous department study; however, it will also include some alterations to provide more stability to the structure and safety during the test. The design can be adjusted for an SGCMG, DGCMG, and VSCMG without changing hardware configurations. A microcontroller will be applied to the testbed to send pulse-width modulation signals (PWM) directly to the two servos and ESCs. With an IMU attached to the testbed, data on quaternions can be fed back to the microcontroller, allowing a closed-loop feedback. Using the integration development environment (IDE) software Platform IO, a code will be used to conduct a test to ensure that the hardware operates properly before implementing a control. Designing the controller can be manually coded using C++, but it will not be used as the primary step. An interface between Simulink and the microcontroller can be established, allowing students to design their controllers similarly to how they would be approached in our control courses. After the design has been completed, a test will be conducted to verify the feasibility and performance on an air-bearing platform, allowing for one degree of freedom. Comparisons between the mathematical model and the test will be analyzed.

2. Modeling Dynamics

2.1 System Description

The design of the CMG is based on a design proposed by the advisor, Dr. Yawo Ezunkpe, and inspired by various existing CMGs mentioned in works of literature. As shown in the schematic in Figure 2.1, it adapts the configuration to the DGCMG that offers two rotatable gimbal bodies, outer and inner, for multiple axes to be controlled. The outer gimbal is mounted relative to the z-axis of the body frame and is freely allowed to rotate about that axis. The inner gimbal is rotatable about the y-axis and is attached to the outer gimbal and flywheels. Each gimbal contains a servo to drive the gimbal position represented with the notations γ_1 and γ_2. The selected servos are designed to rotate with 270 degrees of freedom, but will be limited to a 180 degrees range with a 9.52 rad/s gimbal rate. Additionally, the proposed design uses two symmetrical flywheels to allow for future testing as reaction wheels and redundancy, and the flywheels can operate at lower speeds by increasing the mass and splitting the angular momentum between two flywheel components. Each flywheel is attached to motors mounted on the x-axis of the inner gimbal reference frame. Through this configuration, controlling either gimbal position will allow for a desirable gyroscopic torque to be generated and observed.

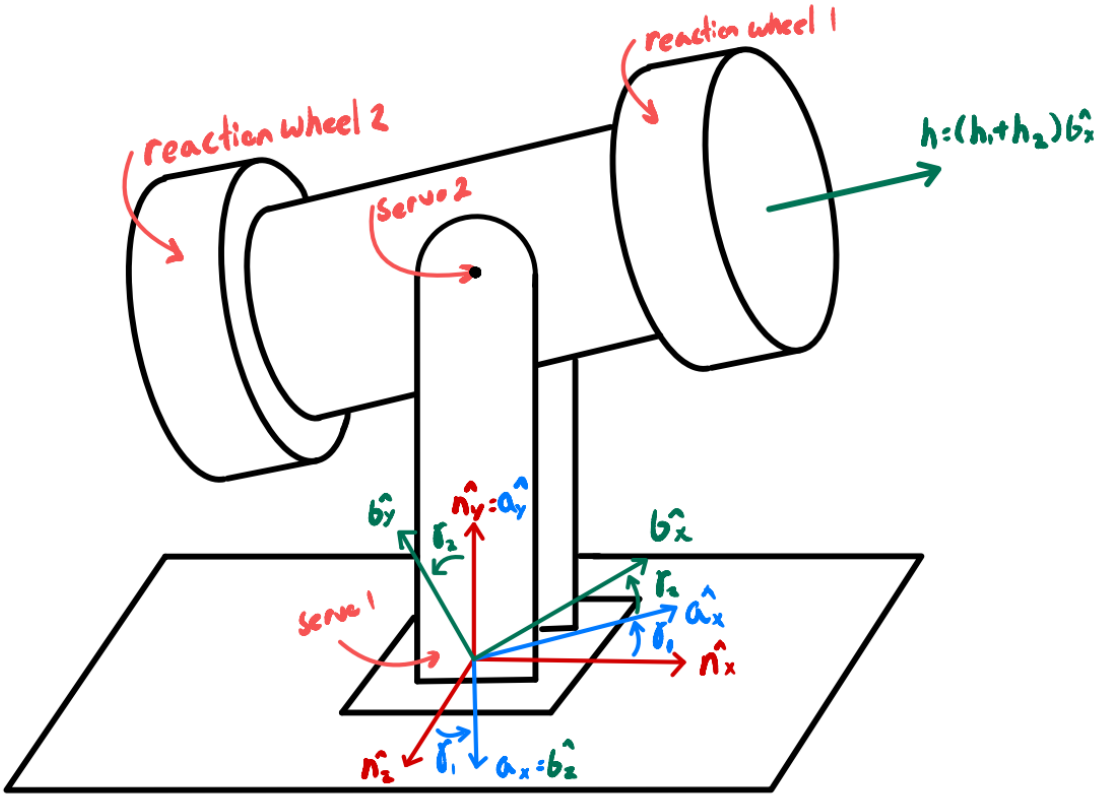

Figure 2.1: Control moment gyroscope illustration of reference frames

The spacecraft will follow a local vertical local horizontal (LVLH) frame, with its origin centered on the celestial body Earth and aligned with the orbital plane. As seen in Figure 2.2, attitude angles can be represented as roll, pitch, and yaw (ϕ,θ,ψ), which are about the x,y, and z axes, respectively. A linearized form to translate the Euler body angular velocity to the attitude angular velocities can be seen in Equation 2.1 provided by Julio Zorita in *Dynamics of Small with Gravity Gradient Attitude Control* [8]. This relationship depends on the mean motion (n) seen in Equation 2.2, where μ_e is the standard gravitational parameter and r is the radius from Earth. Note that the attitude angles equation is linearized and applies a small approximation; thus, the attitude dynamics are accurate about the equilibrium point and range in small attitude angles.

$$\begin{aligned} \dot{\phi} - n\psi &= \omega_1 \\ \dot{\theta} - n &= \omega_2 \\ \dot{\psi} - n\phi &= \omega_3 \end{aligned} \tag{2.1}$$

$$n = \sqrt{\frac{\mu_e}{r^3}} \tag{2.2}$$

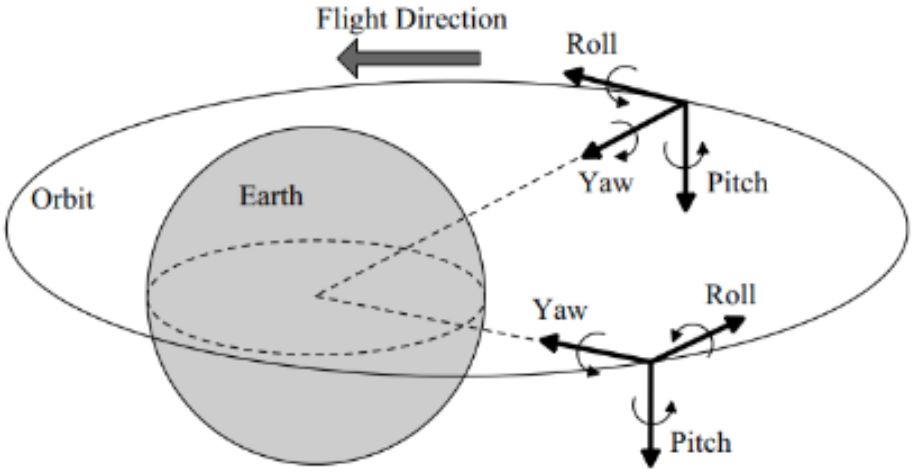

Figure 2.2: Local-vertical local-horizontal attitude reference frame [7]

2.2 Model Assumptions

For our models, Table 2.1 will separate the assumptions for our dynamics of the satellite attitude and the CMG test bench models into separate sections. Spacecraft dynamics will assume conditions at LEO above Earth. As the situation will look at attitudes above 400 km, the significant effect of aerodynamic forces is minimal and will be negligible in the spacecraft model [8].

Table 2.1: Assumptions listed for both the spacecraft attitude in LEO and the CMG testbed

Dynamics of Spacecraft Attitude and CMG	CMG Testbed Dynamics
• Uniformly distributed mass rigid-body of the spacecraft • Gravity gradient in LEO • Neglecting solar and aerodynamic forces for the given attitude • Low-Earth Orbit Conditions • Assume a circular orbit around Earth • The principal axes of the spacecraft constant • Referencing to Local Vertical Local Horizontal (LVLH) • Neglectable small torque produced by the servos • For CMG, will assume a constant angular velocity • Flywheel moment of inertia about the y and z axis will be neglected	• Testbed is limited to rotating only about the y-axis; thus, the rotations about x and z are zero. (ω_1,$\omega_2 = 0$) • Neglectable friction about the testbed y-axis with the use of an air bearing or a possible ball-bearing • Neglectable solar and aerodynamic forces for the given attitude • Neglectable small torque produced by the servos • For CMG, will assume a constant angular velocity • Flywheel moment of inertia about the y and z axes will be neglected

2.3 Rotational Transformations

Two rotational cosine matrices transform the flywheel body frame (b) to the gimbal frame (a) to the stationary reference frame (n) about the origin O. These rotations are related to the gimbal angles γ_1 and γ_2, which will be driven by the servo. The rotational cosine matrices are seen in Equations 2.3 and 2.4, in which the first equation rotates about the y-axis and the second equation rotates about the z-axis. Applying a multiplication matrix to the two cosine matrices can allow direct conversion from b-frame to n-frame. This multiplication matrix can be seen in Equation 2.5.

$$R_y(\gamma_1) = \begin{bmatrix} \cos\gamma_1 & 0 & -\sin\gamma_1 \\ 0 & 1 & 0 \\ \sin\gamma_1 & 0 & \cos\gamma_1 \end{bmatrix} \tag{2.3}$$

$$R_z(\gamma_2) = \begin{bmatrix} \cos\gamma_2 & \sin\gamma_2 & 0 \\ -\sin\gamma_2 & \cos\gamma_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

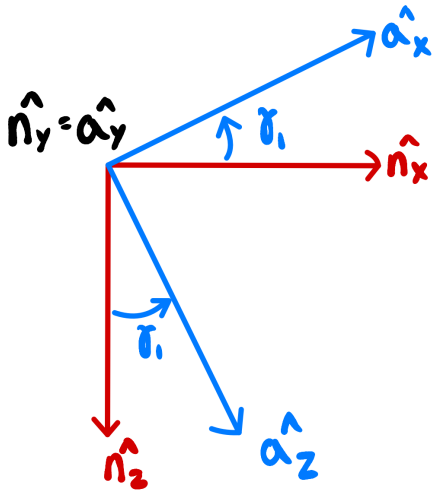

Figure 2.3: Rotation about $\hat{n_y} = \hat{a_y}$

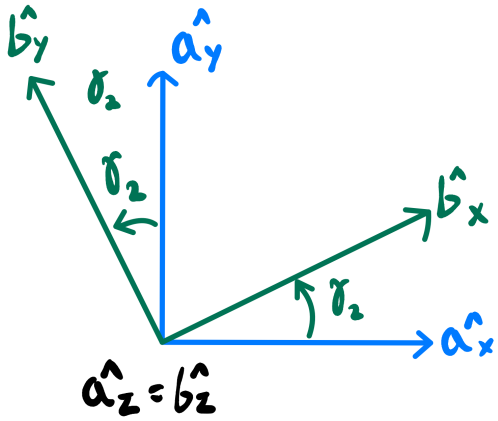

Figure 2.4: Rotation about $\hat{a_z} = \hat{b_z}$

$$R(\gamma_1, \gamma_2) = R_z(\gamma_2)R_y(\gamma_1) = \begin{bmatrix} \cos\gamma_2\cos\gamma_1 & \sin\gamma_2 & -\cos\gamma_2\sin\gamma_1 \\ -\sin\gamma_2\cos\gamma_1 & \cos\gamma_2 & \sin\gamma_2\sin\gamma_1 \\ \sin\gamma_1 & 0 & \cos\gamma_1 \end{bmatrix} \tag{2.5}$$

2.4 Gravity Gradient Torque

Gravity gradient disturbances are usually accounted for in attitudes below 1000 km for LEO [8]. Torque is caused by the distributed mass of the spacecraft and gravitational attraction from the planet it orbits. As some parts of the spacecraft are at slightly different altitudes, the gravitational attraction force varies along the spacecraft and thus results in a torque about the center of mass [8]. Deriving the gravity gradient was referenced with Zorita as seen in Equation 2.6, where each row corresponds to the attitude angles respectively [8]. Understand that the solution is accounted for the LVLH reference frame, simplified with small angle approximation resulting in a negligible yaw motion, and about the principal axes of the spacecraft.

$$\vec{T}_{ext} = \begin{bmatrix} -3n^2(Iy - Iz)\phi \\ 3n^2(Iz - Ix)\theta \\ 0 \end{bmatrix} \tag{2.6}$$

2.5 CMG Quasi-LPV Modeling

A linear time-invariant (LTI) system is limited to capturing the operating ranges of the system as gimbal angles change; therefore, it may not be desired to understand the stability and performance of the system. An LPV model introduces a scheduling parameter function p(t), enabling the representation of nonlinear dynamics to be embedded across various linear systems that are scheduled through the parameter function. In some instances, the parameter function can endogenously depend on the state variables (x) and the input signal (u); therefore, it is a special case of an LPV model known as quasi-LPV. Depending on the dynamics of the CMG, the device may be classified as a quasi-LPV model where the state matrix A and input matrix B change with respect to the gimbal angles. The expression for the quasi-LPV model is represented in a first-order state-space Equation 2.7 such that state space matrix A, input matrix B, output matrix C, and feed-forward matrix D may depend on a p(x,u,t). The parameter function p(x,u,t) may be a matrix in which each entry is bound by given known conditions.

$$\begin{aligned} \dot{x} &= A(\rho)x(t) + B(\rho)u(t) \\ y &= C(\rho)x(t) + D(\rho)u(t) \end{aligned} \tag{2.7}$$

$$\begin{aligned} \rho_{\gamma_1}(x,u,t) &= [\rho_{\gamma_1,low}, \rho_{\gamma_1}, \rho_{\gamma_2,up}] \\ \rho_{\gamma_2}(x,u,t) &= [\rho_{\gamma_1,low}, \rho_{\gamma_2}, \rho_{\gamma_2,up}] \end{aligned} \tag{2.8}$$

2.6 Deriving Equation of Motion

The dynamic of the model can be described through equations of motion that can be derived using Newton's second law for rotation, as seen in 2.9. Based on the characteristics previously described for the CMG and referencing the Figure 2.1, the EOM can be adjusted to represent the model of a DVCMG testbed separately or DVCMG with the spacecraft dynamics. A full step deviation will be included in Appendix A.

$$\frac{d}{dt}\vec{H}_o = \vec{M}_o \tag{2.9}$$

Expanding the left-hand side (LHS), a expression that describes the time derivative of angular momentum can be split into the rate of angular momentum of the body $\dot{\vec{H}}_s$ and angular momentum motion about the body $\vec{\omega}^{\times} \times \vec{H}_s$. H_s corresponds to the total angular momentum of the spacecraft

relative to the spacecraft body, mathematically described in Equation 2.11. The right-hand side represents the sum of external torques T_{ext} experienced by the system. External torques that can be present within the dynamics of the spacecraft include the gravity gradient torque.

$$\dot{\vec{H}}_s + \vec{\omega}^{\times} \times \vec{H}_s = \vec{T}_{ext} \tag{2.10}$$

$$\vec{H}_s = \vec{\vec{I}} \times \vec{\omega} + \vec{h} \tag{2.11}$$

Substituting the Equation 2.11 into Equation 2.10, the result provided below is an expanded form of Newton's second law dynamics of a rigid body and incorporates the effects of angular momentum of the CMGs flywheel $\vec{h}$.

$$(\vec{\vec{I}} * \dot{\vec{\omega}} + \dot{\vec{h}}) + \vec{\omega}^{\times} \times (\vec{\vec{I}} * \vec{\omega} + \vec{h}) = \vec{T}_{ext} \tag{2.12}$$

The angular velocity ω is expressed in the body reference frame, and ω_x is the skew-symmetric matrix $\omega^{\times}$ known as the angular velocity tensor provided in Equation 2.13. $\vec{\vec{I}}$ is the moment of inertia tensor about the principal axes provided as a matrix in Equation 2.14, described as a diagonal tensor.

$$\vec{\omega}^{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.13}$$

$$\vec{\vec{I}} = \begin{bmatrix} Ix & 0 & 0 \\ 0 & Iy & 0 \\ 0 & 0 & Iz \end{bmatrix} \tag{2.14}$$

The angular momentum $\vec{h}$ of two flywheels aligned in the same body-frame x-axis can be described in Equation 2.15. Applying the rotational matrix in Equation 2.3 will help define the momentum of the internal frame. To solve the first derivative of the angular moment of the flywheel, a relationship can be developed that incorporates the Jacobian matrix G(γ_1,γ_2) about the x-axis in the body frame with respect to the gimbal rate angles provided by the servo. As the system inherits a single DGCMG configuration, the Jacobian matrix becomes a 3x2 matrix in terms of γ_1 and γ_2 seen in Equation 2.18.

$$\vec{h} = h_1 + h_2 \tag{2.15}$$

$$\vec{h} = (h_1 + h_2) \begin{bmatrix} \cos\gamma_2 * \cos\gamma_1 \\ \sin\gamma_2 \\ -\cos\gamma_2 * \sin\gamma_1 \end{bmatrix} \tag{2.16}$$

$$\dot{\vec{h}}(\gamma) = \frac{dh(\gamma_{1,2})}{dt} = \frac{dh(\gamma_{1,2})}{d\gamma} * \frac{d\gamma}{dt} = G(\gamma_1, \gamma_2) * \dot{\gamma} \tag{2.17}$$

$$G(\gamma) = \begin{bmatrix} \frac{\partial f(\gamma)}{\partial \gamma_1} & \frac{f(\gamma)}{\partial \gamma_2} \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} -\sin\gamma_1 * \cos\gamma_2 & -\cos\gamma_1 * \sin\gamma_2 \\ \sin\gamma_2 & \cos\gamma_2 \\ -\cos\gamma_1 * \cos\gamma_2 & \sin\gamma_1 * \sin\gamma_2 \end{bmatrix} \tag{2.18}$$

Following the steps in Appendix A, the governing equations will derive the Equation 2.19 seen below. A first-order nonlinear equation of w concerning the body rotating reference frame with 3 degrees of freedom. Note that a section of the equation represents the Euler rigid-body dynamics, while the remaining terms can be viewed as an extension representing the dynamics of the DGCMG attached to the body. From this point, two independent equations of motion will be derived that tackle the testbed dynamics and the spacecraft attitude dynamics. Equation 2.19 will be used as the governing equation for the two EOM derived later in sections 2.6.2 and 2.6.3.

$$\begin{bmatrix} I_x\dot{\omega}_x \\ I_y\dot{\omega}_y \\ I_z\dot{\omega}_z \end{bmatrix} = \begin{bmatrix} (I_z - I_y)\,\omega_y\omega_z \\ (I_x - I_z)\,\omega_z\omega_x \\ (I_y - I_x)\,\omega_y\omega_x \end{bmatrix} - (h_1 + h_2) \begin{bmatrix} -\sin\gamma_2\omega_z - \cos\gamma_2\sin\gamma_1\omega_y \\ \cos\gamma_2\cos\gamma_1\omega_z - \cos\gamma_2\sin\gamma_1\omega_x \\ -\cos\gamma_2\cos\gamma_1\omega_y + \sin\gamma_2\omega_x \end{bmatrix} - \ldots$$

$$\ldots (h_1 + h_2) \begin{bmatrix} -\sin\gamma_1\cos\gamma_2 & -\cos\gamma_1\sin\gamma_2 \\ \sin\gamma_2 & \cos\gamma_2 \\ -\cos\gamma_1\cos\gamma_2 & \sin\gamma_1\sin\gamma_2 \end{bmatrix} \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix} + \vec{T}_{ext}$$

2.6.1 Low Earth Orbit (LEO) Equation of Motion

Continuing with Equation 2.19, the external torque is substituted for the gravity gradient Equation 2.6 provided in section 2.4. Transformations of the angular velocity about the body frame to the attitude angles were applied with LVLH transformation in Equation 2.1. Distributing some of the terms would result in the equation of motion for the attitude dynamics of a spacecraft at LEO with a CMG, as seen in Equation 2.19.

$$\begin{bmatrix} I_x\ddot{\phi} \\ I_y\ddot{\theta} \\ I_z\ddot{\psi} \end{bmatrix} = \begin{bmatrix} n\left(I_x - I_y + I_z\right)\dot{\psi} - 4n^2\left(I_y - I_z\right)\phi \\ 3n^2\left(I_z - I_x\right)\theta \\ n\left(I_x + I_y - I_z\right)\dot{\phi} + n^2\left(I_x - I_y\right)\psi \end{bmatrix} + \ldots$$

$$\ldots (h_1 + h_2) \begin{bmatrix} -\sin\gamma_2\dot{\psi} - \sin\gamma_2 n\phi - \dot{\theta}\cos\gamma_2\sin\gamma_1 + n\cos\gamma_2\sin\gamma_1 \\ \cos\gamma_2\cos\gamma_1\dot{\psi} + \cos\gamma_2\cos\gamma_1 n\phi + \cos\gamma_2\sin\gamma_1\dot{\phi} - \cos\gamma_2\sin\gamma_1 n\psi \\ \sin\gamma_2\dot{\phi} - \sin\gamma_2 n\psi - \cos\gamma_2\cos\gamma_1\dot{\theta} + \cos\gamma_2\cos\gamma_1 n \end{bmatrix} - \ldots$$

$$\ldots (h_1 + h_2) \begin{bmatrix} -\sin\gamma_1\cos\gamma_2 & -\cos\gamma_1\sin\gamma_2 \\ \sin\gamma_2 & \cos\gamma_2 \\ -\cos\gamma_1\cos\gamma_2 & \sin\gamma_1\sin\gamma_2 \end{bmatrix} \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix}$$

The EOM obtained can be placed into the LPV state-space model in which the state variables can be written as $x = [\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ and the input variables can be written as $u = [\dot{\gamma}_1, \dot{\gamma}_2]^T$. By separating the state variables and input variables from the equation, we can obtain the state matrix A and input matrix B used for the state equation. A diagonal matrix is used to output all state variables for the output matrix C. A feed-forward matrix D is set to zero in this case.

$$A(\rho) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ a_{41} & 0 & 0 & 0 & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & a_{56} \\ 0 & 0 & a_{63} & a_{64} & a_{65} & 0 \end{bmatrix} \tag{2.19}$$

$$B(\rho) = -(h_1 + h_2) \begin{bmatrix} \frac{-s_{\rho_1} c_{\rho_2}}{Ix} & \frac{-c_{\rho_1} s_{\rho_2}}{Ix} \\ \frac{s_{\rho_2}}{Iy} & \frac{c_{\rho_2}}{Iy} \\ \frac{-c_{\rho_1} c_{\rho_2}}{Iz} & \frac{s_{\rho_1} s_{\rho_2}}{Iz} \end{bmatrix} \tag{2.20}$$

$$C(\rho) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.21}$$

$$D(\rho) = 0 \tag{2.22}$$

2.6.2 Deriving Equation of Motion for CMG Testbed

Shifting over to the CMG testbed, applying the assumptions and conditions provided in the table would result in the equation of motion for the testbed. The angular velocity and acceleration are zero about the x-axis and z-axis because the system is bound to the y-axis body rotation. Angular velocity is essentially removed, leaving behind the actuation of the CMG and external torque experienced by the system. $\omega_1, \omega_2, \dot{\omega}_1, \dot{\omega}_2 = 0$ was applied to 2.19 resulting in 2.23 being obtained. $\vec{T}_{ext} = 0$ was assumed for this model to analyze how DGCMG affects the testbed directly. Incorporating $\vec{T}_{ext}$ as friction or a damping force between the testbed is also a reasonable assumption to enhance this model.

$$I_y \ddot{\theta} = -(h_1 + h_2)(s_{\rho_2}\dot{\gamma}_1 + c_{\rho_2}\dot{\gamma}_2) + \vec{T}_{ext} \tag{2.23}$$

Terms such as yaw-heading are later referred to as the inertial measuring unit (IMU) portion of the report. For the convention of terminology, pitch (θ) will change to yaw (ψ). In addition, the y-axis will change to the z-axis.

$$I_z \ddot{\psi} = -(h_1 + h_2)(s_{\rho_2}\dot{\gamma}_1 + c_{\rho_2}\dot{\gamma}_2) + \vec{T}_{ext} \tag{2.24}$$

$$A(\rho) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{2.25}$$

$$B(\rho) = -\begin{bmatrix} 0 & 0 \\ \frac{(h_1+h_2)s_{\rho_2}}{Iy} & \frac{(h_1+h_2)c_{\rho_2}}{Iy} \end{bmatrix} \tag{2.26}$$

$$C(\rho) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.27}$$

$$D(\rho) = 0 \tag{2.28}$$

3. Stability and Open-Loop Simulation

3.1 Spacecraft Open-Loop Analysis

3.1.1 Stability affected by Gravity Gradient

Gravity gradient torques can cause a spacecraft to be unstable or stable depending on how the spacecraft is designed and configured. With consideration, one can design a spacecraft capable of maintaining or oscillating about a desired equilibrium point without the need for active controls [8]. Linear stability analysis can be conducted to help further understand this passive stabilization effect. Julia has displayed the mathematical concept for linear stability analysis in multiple literature: *Dynamics of Small Satellites with Gravity Gradient Attitude Control*. Performing this analysis would provide us with the results in Figure 3.1 and Equation 3.3. Figure 3.1 provides a visualization of the gravity gradient stability region in which the non-actuated spacecraft should be designed around to achieve stability. The stability regions in the figure are areas that satisfy all three criteria in Equation 3.1 where k_1 and k_2 are terms related to the moment of inertia of the spacecraft in Equation 3.1 [8].

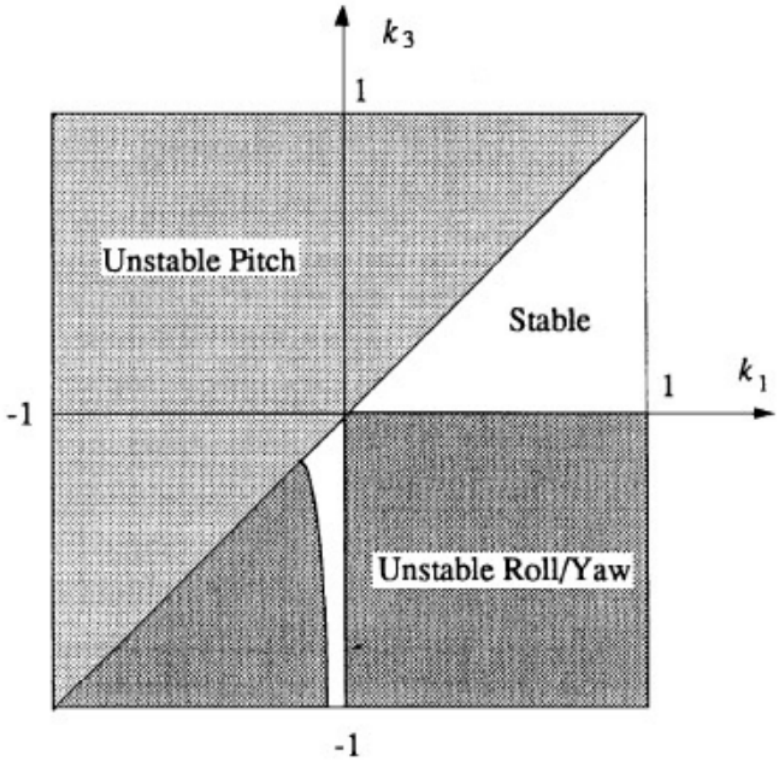

Figure 3.1: Gravity gradient stability regions [8]

$$\begin{aligned} k_1 k_3 &> 0 \\ 1 + 3k_1 + k_1 k_3 &> 0 \\ (1 + 3k_1 + k_1 k_3) - 16 k_1 k_3 &> 0 \end{aligned} \tag{3.1}$$

$$k_1 = \frac{(I_2 - I_3)}{I_1}, \quad k_2 = \frac{(I_2 - I_1)}{I_3} \tag{3.2}$$

$$\begin{aligned} LagrangeRegion\ Region &: Iy > Ix > Iz \\ DeBra - Delp\ Region &: Ix > Iz > Iy \end{aligned} \tag{3.3}$$

A general rule is based on the arrangement of the moment of inertia to satisfy the conditions for stability seen in Equation 3.3. Two types of stability regions can be categorized:

- Lagrange Region: Spacecraft is spun about its major-axis combined with gravity gradient.
- DeBra-Delp Region: Spacecraft is spun about its minor-axis combined with gravity gradient.

3.1.2 Spacecraft Open-Loop Simulation

For this simulation, a 3U CubeSat structure will be referenced from *Attitude Determination and Control of the CubeSat MIST* by Jiewei Zhou, by utilizing the necessary moment of inertia about the principal axis [9]. The orientation of the spacecraft will be subjected to certain initial conditions provided in Table 3.1. For the initial gimbal positions, these were chosen to be aligned such that the flywheels were chosen along the y-axis/normal to the orbital plane. Referencing orbital conditions for LEO, the altitude of a CubeSat is about 600 km from the surface of Earth [8]. In this case, the orbit of the CubeSat will is assumed to be circular to hold n as a constant value.

The angular momentum of the flywheel will be based on various other known control moment gyroscope designs based on Tensor Tech CMG and MicroSat CMG. In the CubeSat case, the flywheel of the spacecraft can be assumed to be one where the total angular momentum of the spacecraft will be $0.0200\ kg \cdot m^2 s^{-1}$ or treated as two flywheels, each with $0.0100\ kg \cdot m^2 s^{-1}$.

Simulations are performed with MATLAB and Simulink using its model-based design features for state-space modeling. The input signals that control the gimbal rates will be assumed to be zero, thus no actuation is present within the simulation. This is to provide a direct analysis of the additional angular momentum from the flywheel and its effects on the spacecraft. This will be done by conducting two open-loop simulations where the angular momentum of the flywheel h_1 and h_2 are considered to be 0, thus allowing for only the gravity gradient effects to be present. The second simulation will provide the flywheels and gravity gradient effects together.

Table 3.1: Spacecraft parameters and initial conditions [8, 9]

Parameter	Value
Angular Momentum $[h_n]$	$h_1 = 0.0100\ kg \cdot m^2 s^{-1}, h_2 = 0.0100\ kg \cdot m^2 s^{-1}$
Moment of Inertia [I]	$\begin{bmatrix} 0.037\ kg \cdot m^2 & 0 & 0 \\ 0 & 0.051\ kg \cdot m^2 & 0 \\ 0 & 0 & 0.021\ kg \cdot m^2 \end{bmatrix}$
Initial Attitude Position $[(\phi, \theta, \psi)_{initial}]$	$[15^o, 12^o, 8^o]$
Initial Attitude Rates $[(\dot{\phi}, \dot{\theta}, \dot{\psi})_{initial}]$	$[0^o, 0^o, 0^o]$
Gimbal Initial $[\rho_{\gamma_{1(Init)}}, \rho_{\gamma_{2(Init)}}]$	$[0^o, 90^o]$
Gimbal Boundary $[\rho_{\gamma_1,low}, \rho_{\gamma_1,up}][\rho_{\gamma_1,low}, \rho_{\gamma_2,up}]$	$[-90^o, +90^o], [0^o, +180^o]$
Altitude [a]	600 km
Earth Radius [R]	6371 km

Figure 3.2 is the Simulink environment for the open-loop simulation for the spacecraft at LEO orbit. An LPV block is necessary to simulate the varying state-space models to which its parameters are assigned. The parameters are the gimbal angles for the CMG in a 2x1 matrix. These angles can be obtained by placing an integral block with the gimbal angle rates and specifying the initial gimbal angles within the block.

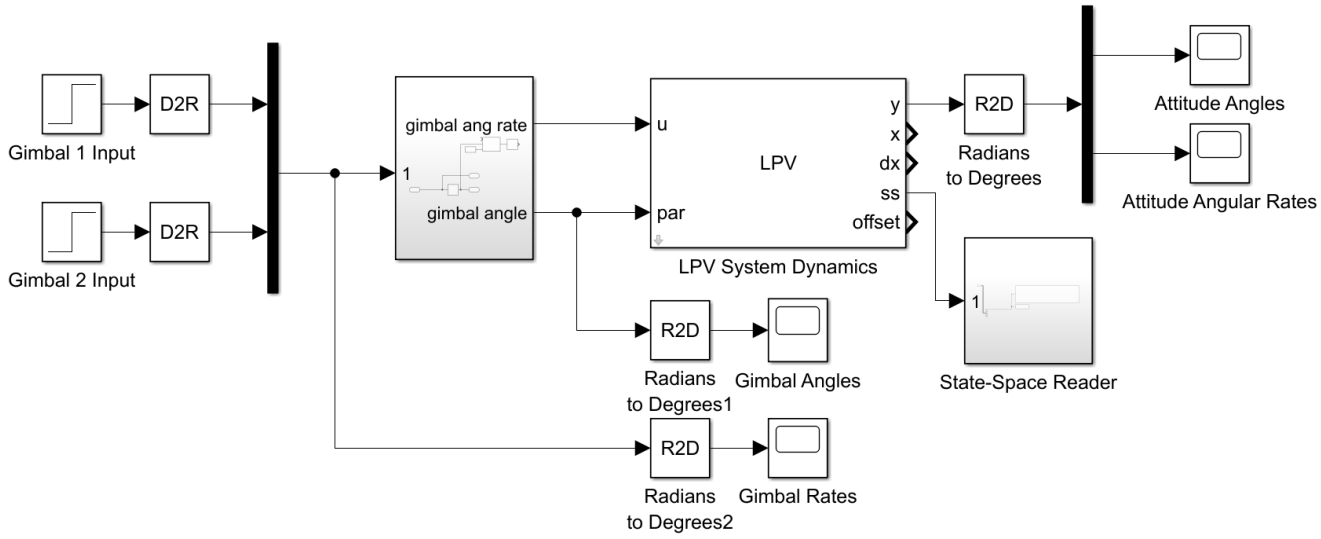

Figure 3.2: Open-loop using LPV system

The system state-space output matrix is an identity matrix; thus, the response of the system will provide all six state variables: the three attitude angles and their corresponding rates. The simulation is run at a finite time that is specified within the MATLAB code used for open-loop analysis provided in Appendix F, combined with Appendix G. It is computationally solved with the automatic solver selection option in the Simulink settings. When run, the solver is automatically selected to use ODE45 for continuous signals.

In Figures 3.3 and 3.4, the response for the gravity gradient simulation for the spacecraft is simulated. Analyzing the attitude angles indicates that the system is seen to be mutually stable and continuously oscillates about the equilibrium point. As $Iy > Ix > Iz$ is true for the given 3U CubeSat, this stable behavior was to be expected and thus satisfies the Lagrangian criteria for the gravity gradient condition. Note how roll and pitch exhibit a typical known oscillatory behavior while yaw performs a unique oscillation that is still considered stable. Similar behavior can be seen within the rates as well.

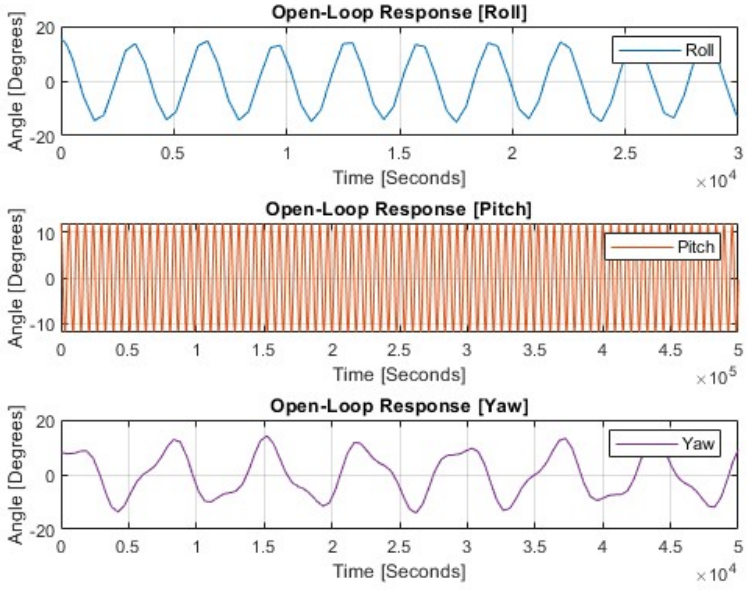

Figure 3.3: Open-loop attitude angle response for spacecraft with gravity gradient only

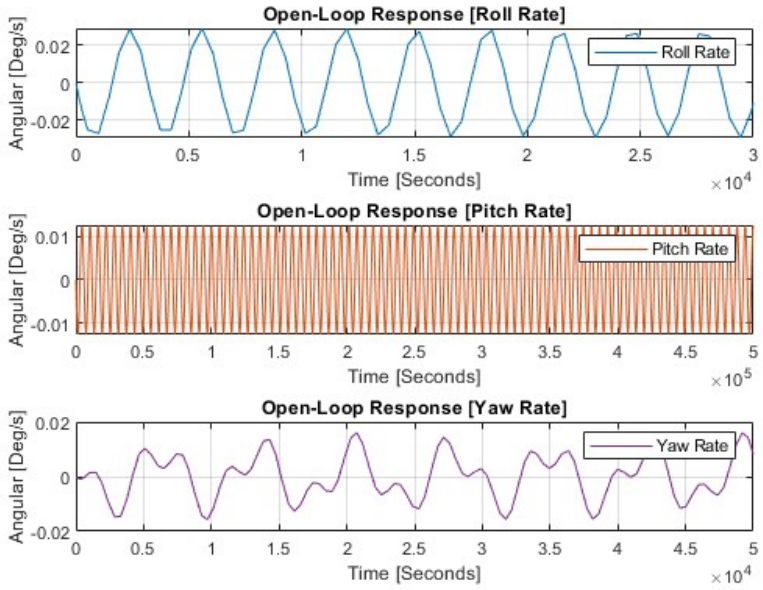

Figure 3.4: Open-loop attitude angle rate response for spacecraft with gravity gradient only

Next, the open-loop for the satellite with CMG and gravity gradient is performed with the same initial conditions as the previous simulation. Figures 3.5 and 3.6 illustrate the responses for this system and thus show a mutually stable response for all six state variables. Compared with the simulation with the gravity gradient only, the yaw angle smoothed out to a typical oscillation with the flywheels present. Furthermore, a slight distinct wobbling effect is inherited from its roll and yaw response as shown in Figure 3.8. This wobbling effect is more clearly indicated by analyzing the attitude rates and can be similar to the nutation experienced by rotating bodies. To prove that the system is marginally stable, a deeper analysis is done by analyzing the poles of the system about the initial gimbal angles of the CMG.

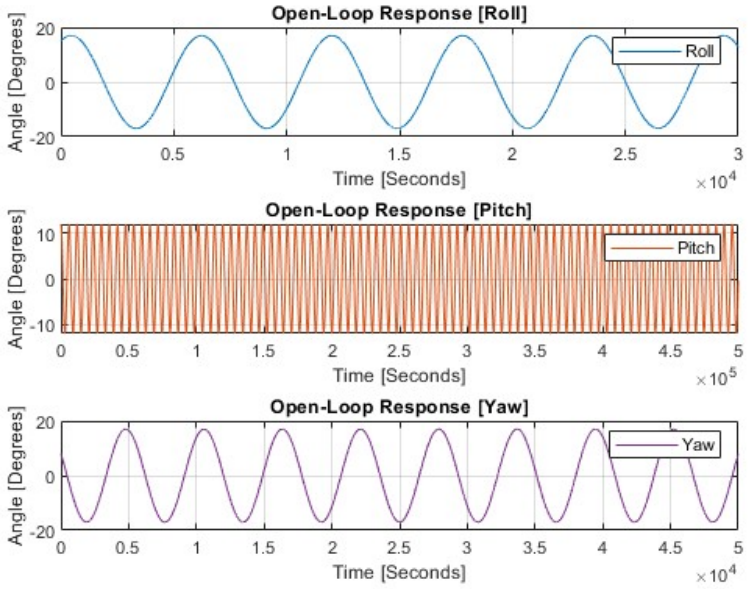

Figure 3.5: Open-Loop attitude angle response for spacecraft DGCMG & gravity gradient

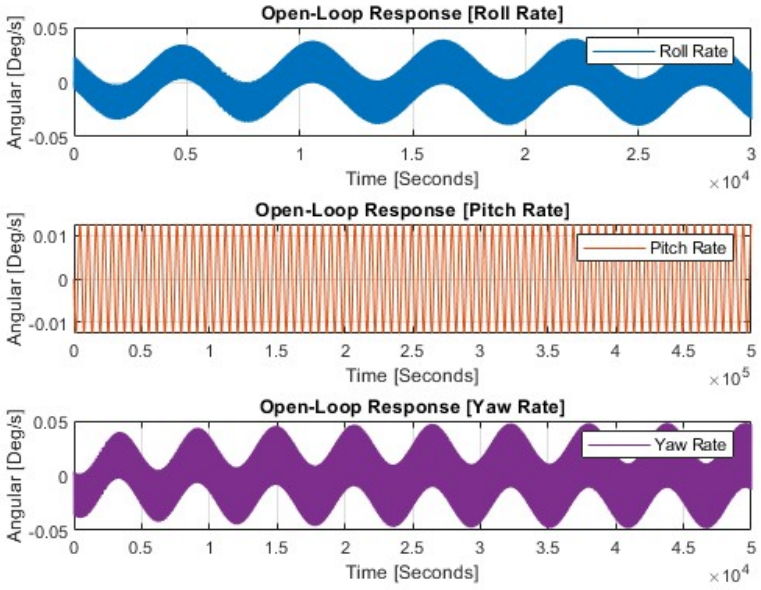

Figure 3.6: Open-Loop attitude angle rate response for spacecraft DGCMG & gravity gradient

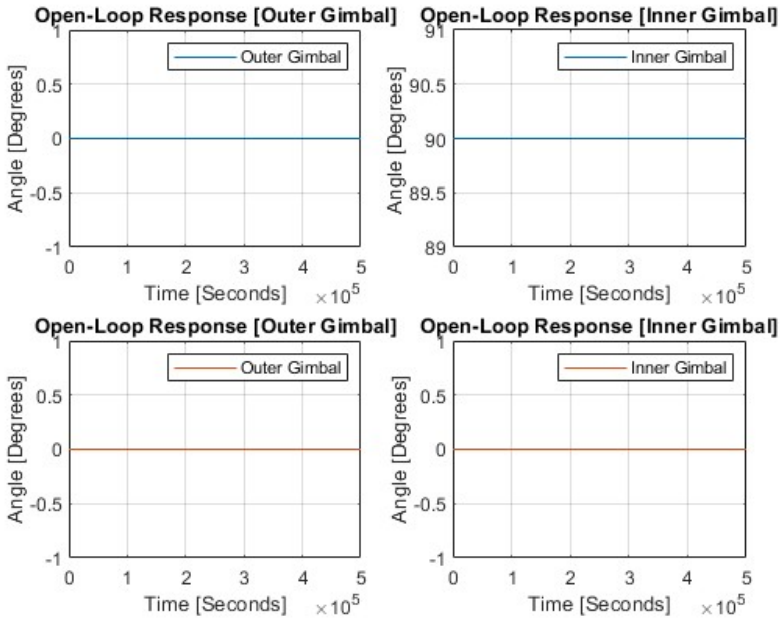

Figure 3.7: Open-Loop gimbal angle for spacecraft DGCMG & gravity gradient

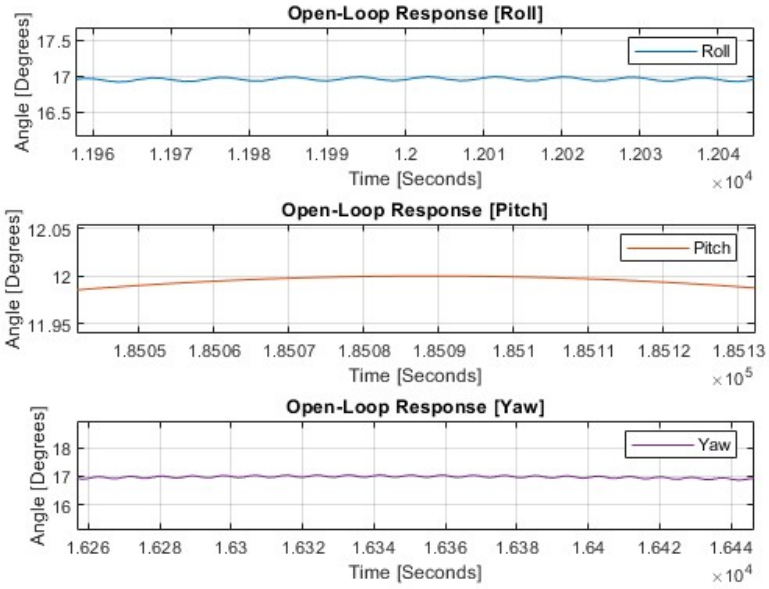

Figure 3.8: Attitude angles zoomed-in to show the wobbling effect in the spacecraft DGCMG & gravity gradient open-loop simulation

3.2 Pole Location Analysis

In linear time-invariant (LTI) systems, pole analysis is used to help determine the stability of the open-loop system. However, the dynamics of spacecraft with CMG are considered to be nonlinear systems, and this is treated by implementing an LPV system that varies across different parameter conditions. Though this method is not entirely intended for the LPV models, pole analysis can still be used to help analyze the stability of a local LPV model by ”freezing” the parameters to specified constant values. Our case assumes the initial conditions for the gimbal angles for the open-loop system. Various gimbal angles can be used to analyze the stability of that given parameter.

For the attitude dynamics of a spacecraft, six poles can be obtained corresponding to each state variable. Poles represent complex eigenvalues for the system given the state-space model. Complex eigenvalues consist of real and imaginary values that are key components in defining the stability and behavior of the system. The fundamentals of linear algebra note that a system is determined to be stable if all its real parts are negative, thus providing an asymptotically stable solution about the equilibrium point. An upper and lower boundary will bind the response of a stable system and will converge to a steady state over time. If a linear system diverges from its steady state, the system is said to be unstable. A list of more linear system behaviors for stability will be listed below.

- If the real parts of the poles are negative, then the system is stable
- If the real parts of the poles are positive, then the system is unstable
- If the real parts of the poles are zero, then the system is considered to be marginally stable
- If imaginary parts exist within the poles, the system will exhilarate oscillations

Poles can be obtained by finding the characteristic polynomial provided in Equation 3.4, which is solved when the equation is set to 0. Where $p(\lambda)$ is the polynomial equation, $A(\rho)$ is the state matrix, λ is the poles of the system, and I is the identity matrix 6x6. For the open-loop analysis, Matlab command $/pole(IC(:,:,point_{center1},point_{center2}))$ was used to computationally used to solve for the poles for the initial conditions. $point_{center}$ is the center location of the grid where the initial conditions of the gimbal lay-on. Note that $/eig(IC(:,:,point_{center1},point_{center2}))$ can achieve the same result.

$$p(\lambda) = det(\lambda I - A(\rho)) \tag{3.4}$$

Table 3.2: Pole locations of two simulations [gravity gradient] and [gravity gradient + CMG]

Poles	CMG + Gravity Gradient	Gravity Gradient
$P_1 =$	$4.8012\text{e}^{-56} + 7.1835e^{-01}i$	$0 + 1.9771\text{e}^{-03}i$
$P_2 =$	$4.8012\text{e}^{-56} - 7.1835e^{-01}i$	$0 - 1.9771\text{e}^{-03}i$
$P_3 =$	$0 + 1.0874\text{e}^{-03}i$	$0 + 8.7511\text{e}^{-04}i$
$P_4 =$	$0 - 1.0874\text{e}^{-03}i$	$0 - 8.7511\text{e}^{-04}i$
$P_5 =$	$3.3959\text{e}^{-72} + 1.0524e^{-03}i$	$0 + 1.0524\text{e}^{-03}i$
$P_6 =$	$3.3959\text{e}^{-72} - 1.0524e^{-03}i$	$0 - 1.9771\text{e}^{-03}i$

A visualization of the poles can be graphed onto a complex plane, with the real and imaginary values placed on separate axes; this graph is known as the pole-zero map. This method is designed to be a graphical representation in classical control theory to help determine pole stability. The pole-zero analysis is mainly for linear time-invariant (LTI) systems and is not intended for LPV systems. However, similar steps in freezing the parameters can help determine the stability of the system. Ranging at various conditions has been done for the simulation with gravity gradient and CMG to determine the stability at various parameters as seen in Figure 3.10. The visualization of the pole-zero map for the LPV model allows a definable layout for the poles, making it condensable to understand rather than analyzing numerical poles directly.

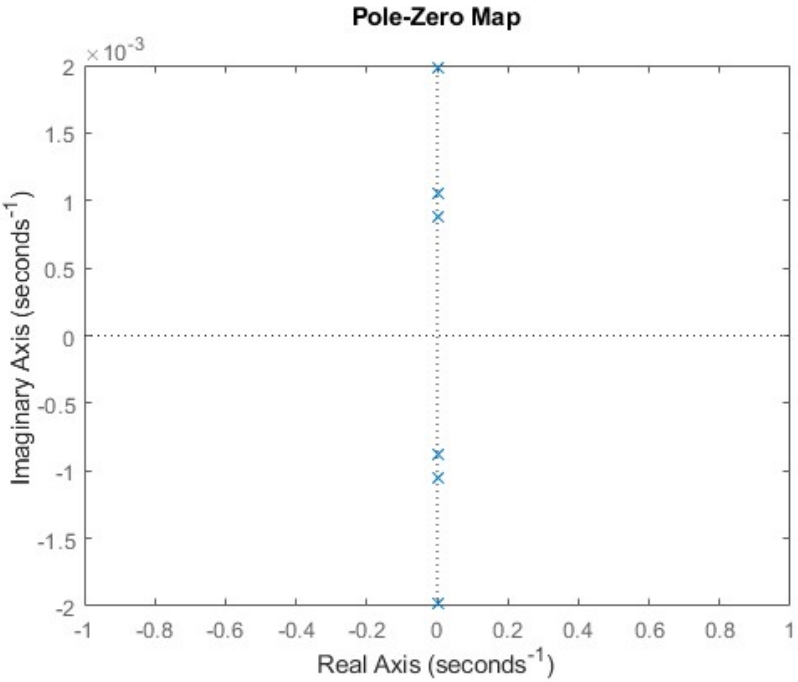

Figure 3.9: Poles of LPV model for satellite with only gravity gradient

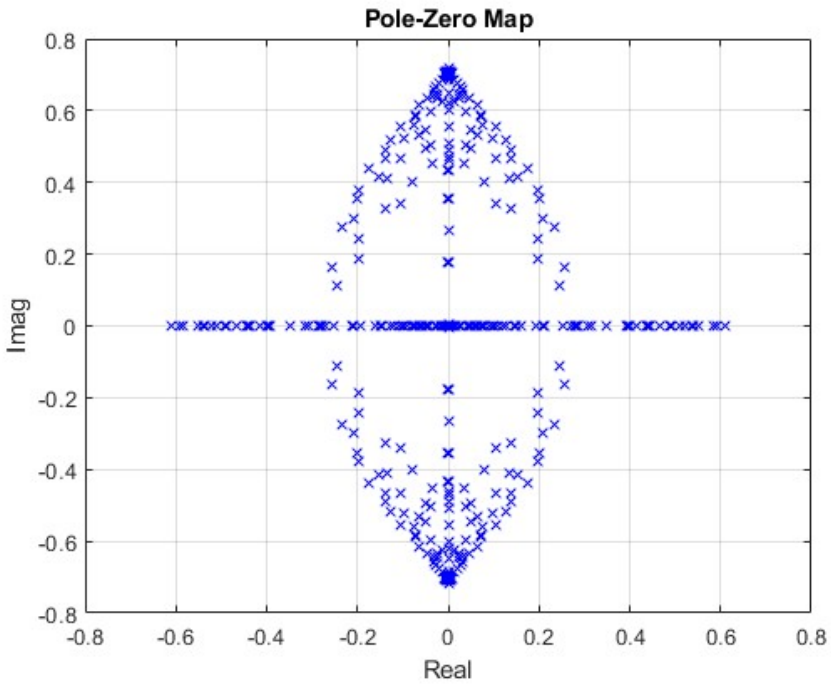

Figure 3.10: Poles of LPV model for satellite with DGCMG & gravity gradient

Analysis of the poles results can be done by comparing Table 3.2. By performing this analysis, stability for the system can be determined for an infinite amount of time without resolving it through solvers and simulation. In the general gravity gradient case on Table 3.2, a clear indication of the poles shows that the system is marginally stable as each of the real parts of the poles is 0. It is expected to have oscillations due to imaginary terms being present. This result is supported by the open-loop response seen in Figures 3.3 and 3.4. In the CMG and gravity gradient case, P1 and P2 are terms that gain a real positive value but in magnitudes that are e^{-56}, which is considered to be significantly neglectable to affect the stability of the system. The imaginary terms are also considered greater than the first case. Therefore, the system is considered to be marginally stable, imparting higher frequencies of oscillation compared to the gravity gradient. This higher frequency oscillation can be associated with the wobbling effect caused by the CMG. Remember that these results are only for the initial conditions for the gimbal rates.

A more in-depth stability analysis at various parameter conditions is conducted using Figure 3.10. The Pole-Zero Map lists 289 possible combinations where the poles would be placed from each LTI system pole. Each LTI corresponds to various gimbal angles of γ_1 and γ_2 between the bounded conditions provided in Table 3.1. Though not directly defined in terms of gimbal angles, stability analysis does indicate that there exist gimbal angle conditions where the LPV model will behave unstable. Thus, for a spacecraft with a CMG and gravity gradient, concerns in the design of an open-loop spacecraft should be considered when using a CMG.

3.3 Testbed Open-Loop Analysis

A process similar to that of the open-loop satellite is performed for the testbed. In this case, we adjust the parameters and initial conditions for the testbed as seen in Figure 3.3. Calculations for the angular momentum of the flywheels is calculated in the same method of the spacecraft. The moment of inertia of the spacecraft is obtained from the CAD model designed within SolidWorks.

In addition, alterations to the input have been made to see how the system is perturbed when an input signal is applied. Two doublet input signals seen in Figure 3.13 were chosen to bring the gimbal rate signal back to rest after torque has been applied. The Simulink model is displayed in Figure 3.11 and was simulated for 50 seconds with ODE45 as the solver.

Table 3.3: Testbed parameters and initial conditions

Parameter	Value
Angular Momentum $[h_n]$	$h_1 = 0.05\ kg \cdot m^2 s^{-1}, h_2 = 0.05\ kg \cdot m^2 s^{-1}$
Moment of Inertia $[I_z]$	$0.02\ kg \cdot m^2$
Initial Yaw Position $[\psi]$	0^o
Initial Yaw Rate $[\dot{\psi}]$	0^o
Gimbal Initial $[\rho_{\gamma_{1(Init)}}, \rho_{\gamma_{2(Init)}}]$	$[0^o, 0^o]$
Gimbal Boundary $[\rho_{\gamma_1}, \rho_{\gamma_2}]$	$[-90^o, +90^o], [-90^o, +90^o]$

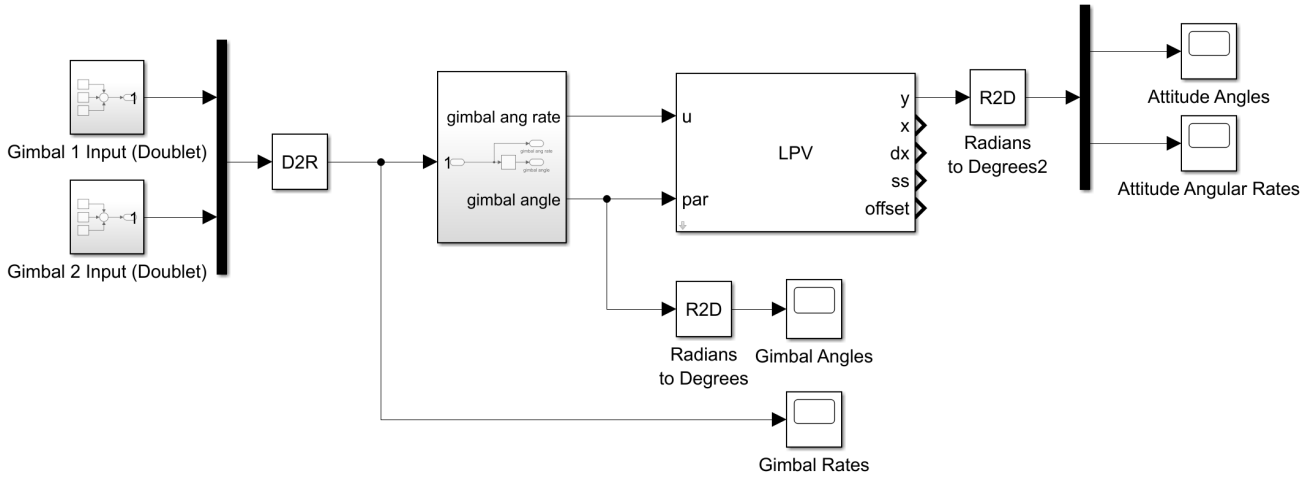

Figure 3.11: Testbed open-loop Simulink LPV model

Figure 3.12 and 3.13 are the responses of the Testbed dynamics. Analysis of the response indicates a marginally stable system that will remain fixed in its position if no forces are applied. Without a doublet input response, the position will continue to stay in motion until the CMG counteracts the motion with an additional external torque. Thus, a control method is required to improve the stability and performance. In addition, a minimal change in the gimbal angle is needed to induce a large angular position change on the testbed. About a 2^o change in both the inner and outer gimbal was capable of changing the position of a 0.02 $kg \cdot m^2$ testbed by -40^o within 10 seconds before reaching a steady state.

For the hardware test, an evaluation incorporating friction or a damping force may be suggested to model more accurately for a testbed system designed for longer runtimes. In addition, a more accurate model for the flywheel angular momentum and testbed moment of inertia will be needed.

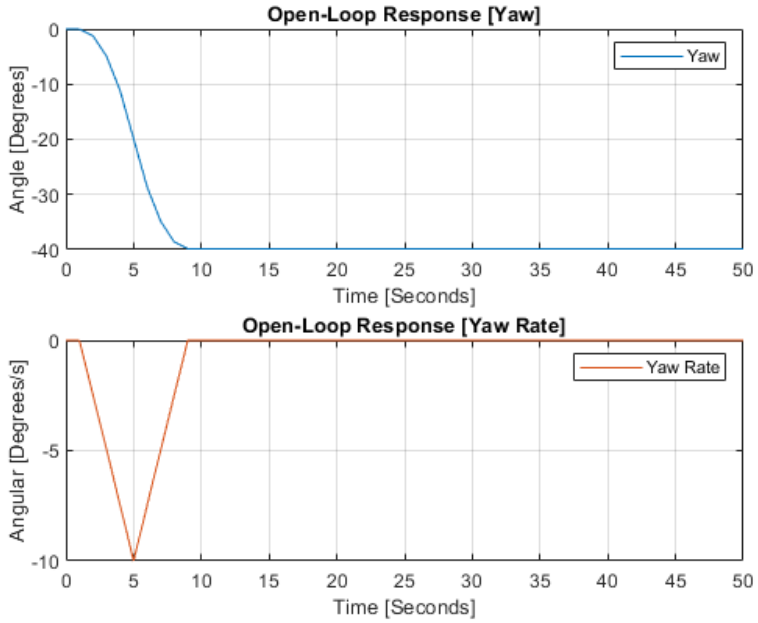

Figure 3.12: Testbed open-loop state response

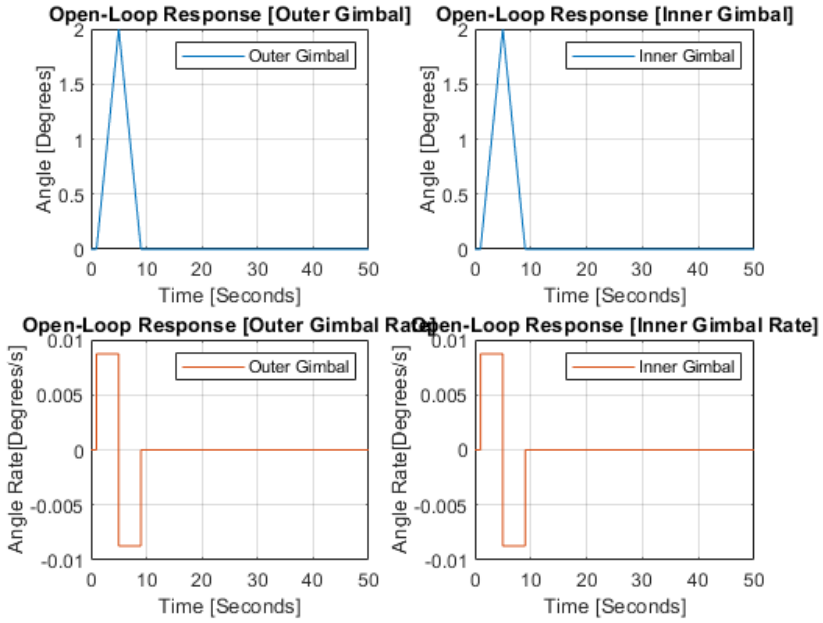

Figure 3.13: Testbed open-loop gimbal input response

Confirmation that the system stability can be supported by looking at the poles of the system. Applying Equation 3.4, results are obtained in Table 3.4. The system consists of all zero poles and is marginally stable.

Furthermore, the poles of the system for any configuration of the testbed will be the same for each LPV model, as the state-space matrix A is independent of the parameter function. Thus, the analysis on the poles in Table 3.4 is sufficient to determine the stability of this system, and a pole-zero map will not be needed for assessing the stability of the LPV model across each parameter.

Table 3.4: Pole locations testbed

Poles	Value
$P_1 =$	$0.0 + 0.0i$
$P_2 =$	$0.0 + 0.0i$

4. Controller Design and Closed-Loop Simulation

4.1 Controllability and Controller Design

4.1.1 Controllability

Before a controller can be implemented for the CMG, it is essential to determine whether the system states are steerable to desirable values. Typically, a controllable linear system can drive the system to the specified states at any given initial condition. Certain control methods, such as linear quadratic regulators, require the system to be controllable to compute the optimal control gains by solving the Riccati equation. Solving for controllability for a linear system can be achieved by implementing the Kalman Controllability Matrix defined in Equation 4.1 and its rank in Equation 4.2. If the matrix rank is equal to the number of states in the system, then the system is said to be fully controllable.

$$Co = \left[B|AB|...|A^{n-1}B\right] \tag{4.1}$$

$$n_{co} = rank(Co) \tag{4.2}$$

For the given CMG LPV model, assumptions were made to determine controllability across parameters that were considered to be frozen. Thus, controllability would be determinable for a single linear time-invariant system within the LPV model. By re-conducting the same procedure across various parameters, confirmation for stability can be determined across the bounded ranges. Therefore, an LPV is controllable for all bounded ranges of ρ as long as the rank of the controllability matrix is equal to the number of states.

Performing the procedure for both models of the CMGs using MATLAB, the rank of the controllability matrix can be determined. The results of the analysis indicate that the systems are fully controllable as the $rank(Co(\rho)) = 6$ for the spacecraft dynamics and $rank(Co(\rho)) = 2$ for the testbed dynamics. Thus, the dynamics of the model can be controlled by the given parameters. A LQR system can be implemented as the controller for the DVCMG.

4.1.2 Gain-Scheduling LQR Controller

LQR is an optimal control technique that uses a state-space representation. It operated by attempting to minimize the performance index seen in Equation 4.3 to balance the energy cost given by the control input and state regulation cost. This is done when the input signal for the system is given as Equation 4.5. The gain matrix K is achievable by solving for a solution P in the algebraic Riccati Equation 4.4 and implementing the solution in Equation 4.6. Matlab was used to compute the optimal gain matrices by using the built-in LQR function $lqr(ICs(:,:,i,l), Q_w, R_w)$.

Tuning the LQR controller is done through the two weighted matrices in Equation 4.3: a positive-semidefinite state weight matrix Q_w and a real symmetric matrix control input weight matrix R_w. Tuning these matrices allows for changing the significances or penalizing certain characteristics of the system. The state weight matrix can alter how the desirable states may deviate. The control input matrix will alter the control effort that the actuator applies to the system.

$$J = \int_0^\infty (x^T Q x + u^T R u) \tag{4.3}$$

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \tag{4.4}$$

$$u(t) = -Kx(t) \tag{4.5}$$

$$K = R^{-1}B^T P \tag{4.6}$$

4.1.3 Designing Linear-Quadratic Regulator for Testbed

LQR is used for linear systems; however, by applying gain-scheduling, it is possible to extend an LQR model to provide stability and performance for the given LPV system of the CMG. By mapping out the gain matrix across the bounded parameters ρ_1 and ρ_2, a selective gain matrix can be adjusted to match the corresponding LPV system depending on the two known gimbal angles of the DGCMG. This is achievable using multiple 2D lookup tables block functions within Simulink to search each element for the gain matrix individually, as seen in Figure 4.2. Substituting the gain-scheduling matrix model into the optimal location of the gain matrix, a gain-scheduling LQR system is designed as shown in Figure 4.1 and 4.9.

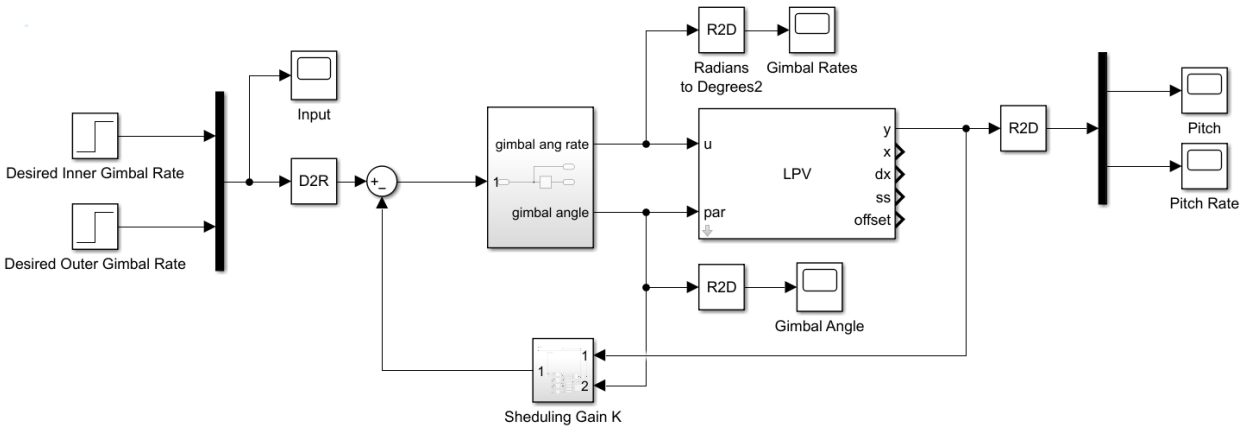

Figure 4.1: Testbed LQR & LPV Simulink model

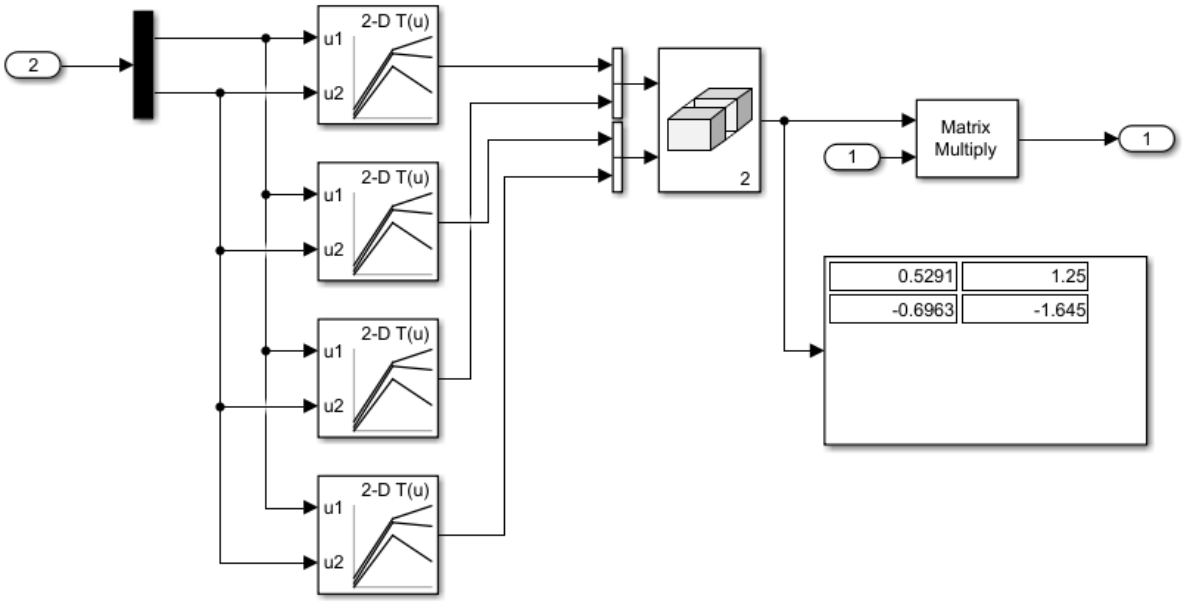

Figure 4.2: Testbed LQR gain-scheduling matrix Simulink model

The testbed control model was computed using Simulink, displayed in Figure 4.1. The desired gimbal rates for both servos were set to 0 deg/s. As the LQR controller is intended to regulate the state variables to the equilibrium points, the system response is expected to drive the pitch back to 0 degrees when achieving a steady state. The stability of the system about the equilibrium point is confirmed in Figure 4.3. Furthermore, the gimbal responses for the system become steady in both pitch and pitch rate. The actuator efforts are not needed for the LQR controller to drive the system to equilibrium.

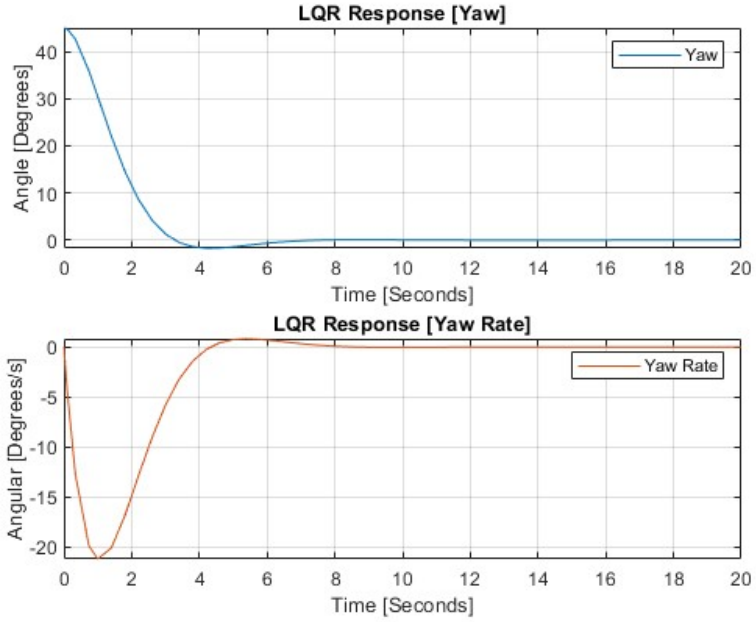

Figure 4.3: Testbed LQR state response

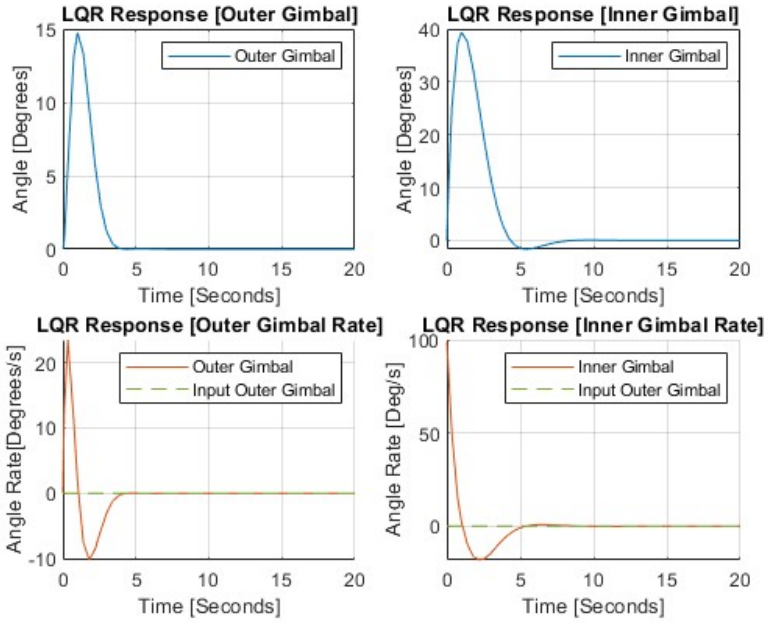

Figure 4.4: Testbed LQR gimbal response

Though the LQR systems presented in Figure 4.1 have achieved stability for the system, limitations are shown in attempts to drive the system to a desirable pitch angle. Alterations to the input signal can be made to help relate the desirable pitch angle to the LQR model. This would allow the system to track desirable pitch angles using the same optimal gain matrices terms solved for the LQR system.

An adjustment to the LQR model for the testbed in Figure 4.5 was implemented to test this controller to achieve desirable trackability for a pitch angle of 20 degrees. Results in Figure 4.7 and 4.8 show the systems and gimbal responses. It can be seen that the system drives to its desired pitch angle of 20 degrees approximately around 12 seconds. Therefore, the improved trackability may be relevant for certain conditions of the LQR single-input multi-output testbed system.

$$u(t) = K_1 r_1 - (K_1 x_1(t) + K_2 x_2(t)) \tag{4.7}$$

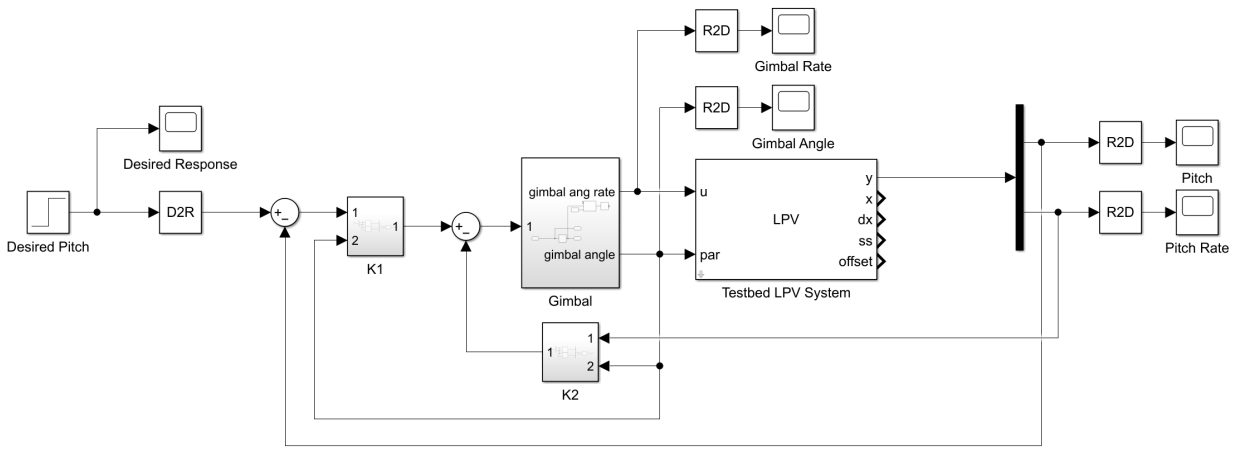

Figure 4.5: Testbed LQR for improved tracking Simulink model

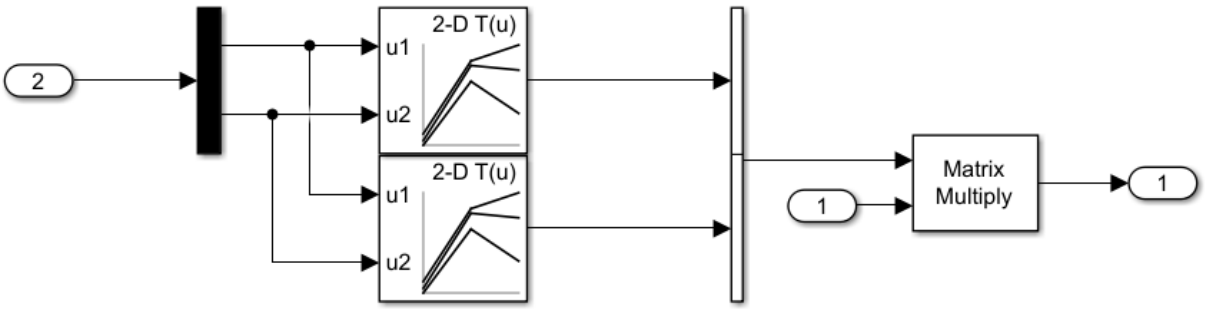

Figure 4.6: Testbed LQR-tracking gain-scheduling matrix Simulink model

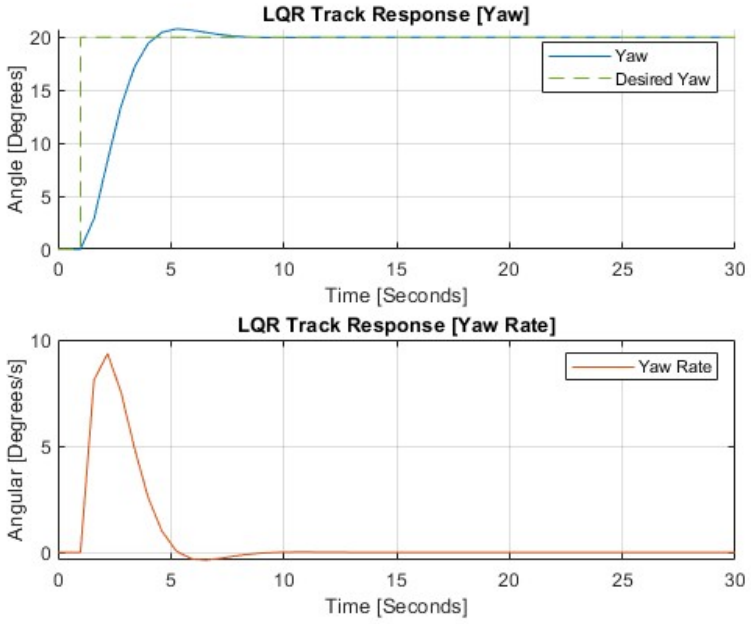

Figure 4.7: Testbed LQR-tracking state response

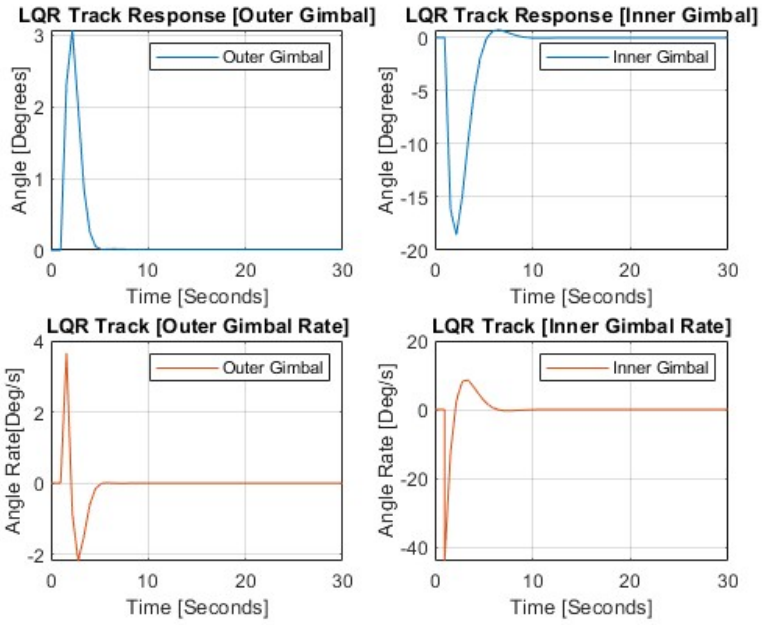

Figure 4.8: Testbed LQR-tracking gimbal input response

4.1.4 Designing Linear-Quadratic Regulator for Spacecraft

The LQR controller was implemented for the LPV model of the spacecraft to regulate the attitude angles and rates back to the equilibrium point at which the LPV model was linearized. Initial conditions used for the LQR controller are provided in Table 4.1. The positions of the gimbals will remain in the same orientation as in Table 3.1.

Running the simulation, the results in Figures 4.10 and 4.11 display the response of the system for 30000 seconds and provide support that the LQR model can help drive a spacecraft to equilibrium with a single DGCMG device. Results indicate that pitch and roll are responsive in stabilizing about the equilibrium points within 10 seconds of the simulation. In contrast, yaw is recognized to stabilize much slower than roll and pitch. It takes approximately 2000 seconds for Yaw to stabilize at its equilibrium.

Table 4.1: Initial conditions for spacecraft LPV model

Parameter	Variable
Initial Attitude Position $[(\phi, \theta, \psi)_{initial}]$	$[10^\circ, 15^\circ, -15^\circ]$
Initial Attitude Rates $[(\dot{\phi}, \dot{\theta}, \dot{\psi})_{initial}]$	$[-15^\circ, 5^\circ, 10^\circ]$

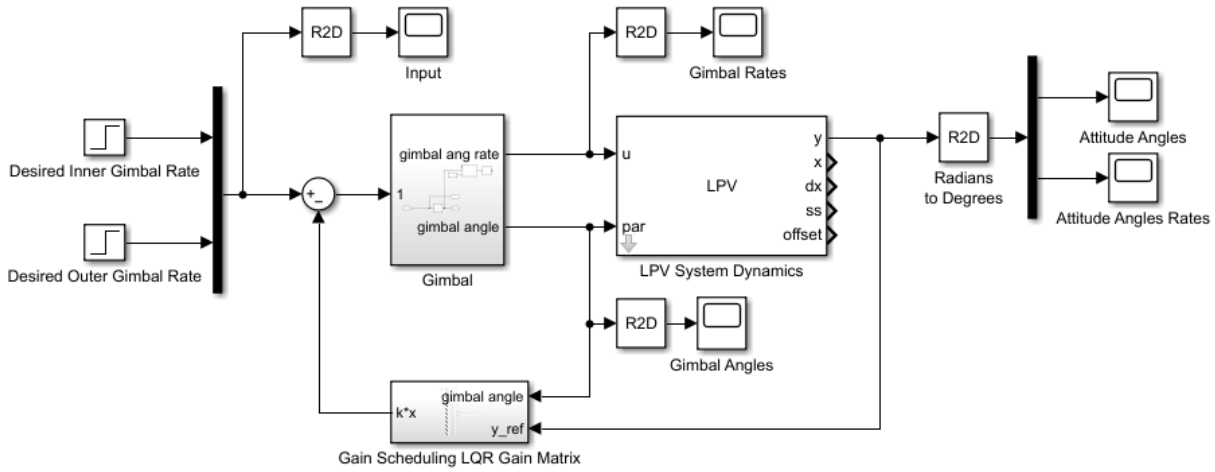

Figure 4.9: Spacecraft LQR Simulink model

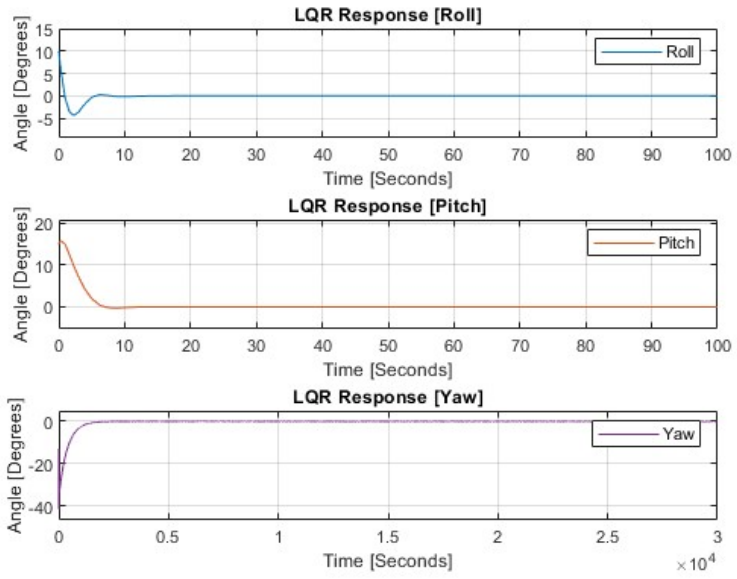

Figure 4.10: Spacecraft LQR attitude response

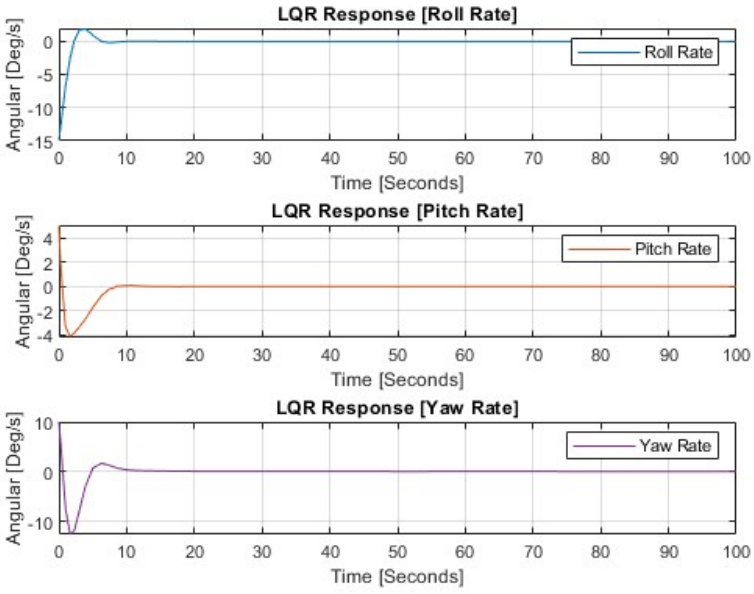

Figure 4.11: Spacecraft LQR attitude rate response

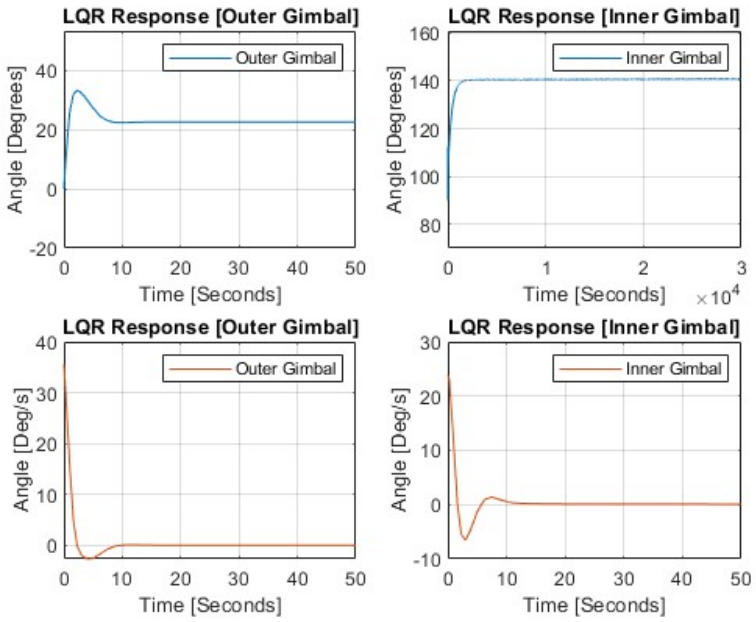

Figure 4.12: Spacecraft LQR gimbal input response

Improving the tracking of the system is done by implementing Equation 4.8. Three separate simulations were done to test whether the controller was capable of tracking roll, pitch, and yaw. Each simulation would test each attitude angle with a response that would be used to hold the spacecraft at 15 degrees for 4000 seconds. The simulation would run for 10000 seconds to ensure the results reach a steady state after the input signal.

$$u(t) = K_1 r_1 + K_2 r_2 + K_3 r_3 - (K_1 x_1(t) + K_2 x_2(t) + K_2 x_2(t) + K_3 x_3(t)) \tag{4.8}$$

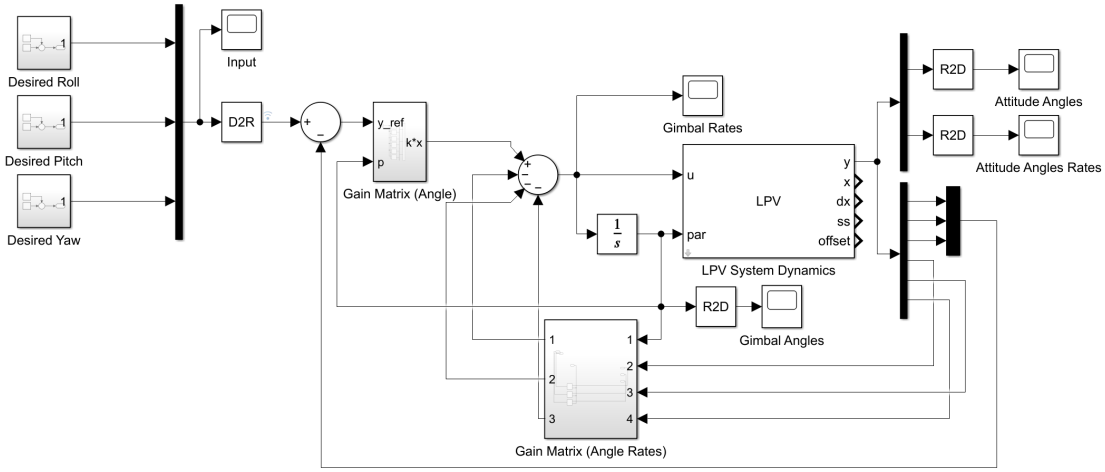

Figure 4.13: Spacecraft LQR-tracking LPV model Simulink

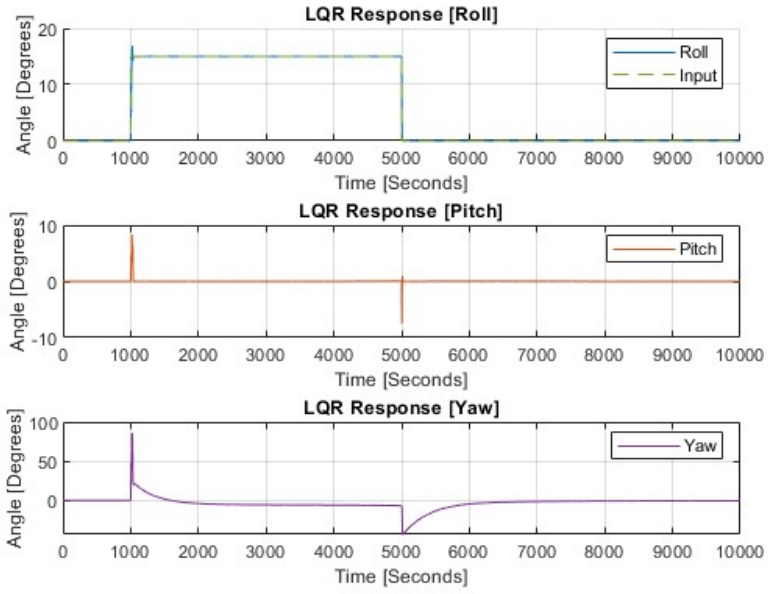

Figure 4.14: Spacecraft LQR-tracking roll response

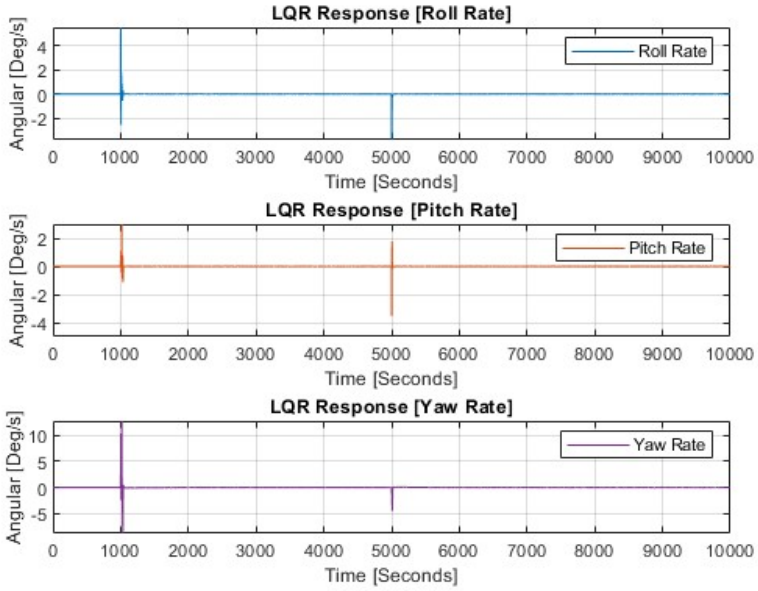

Figure 4.15: Spacecraft LQR-tracking roll rate response

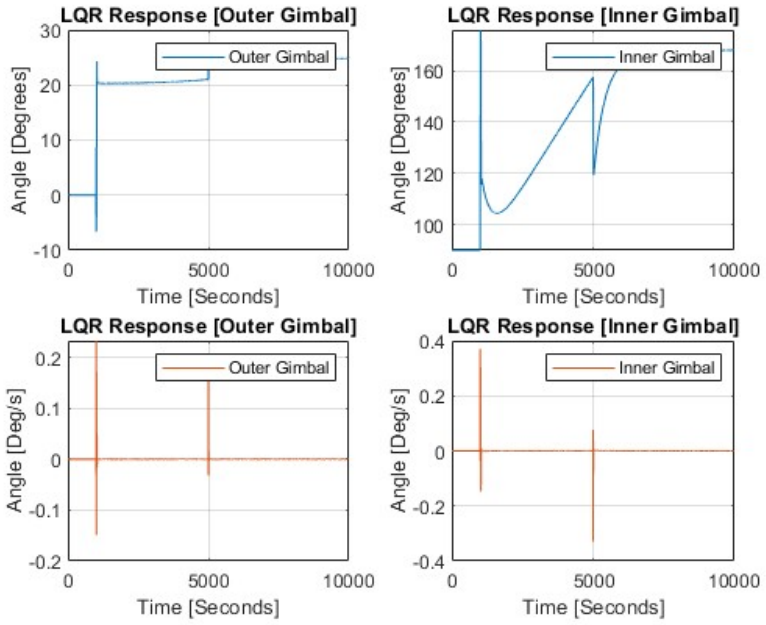

Figure 4.16: Spacecraft LQR-tracking roll gimbal input

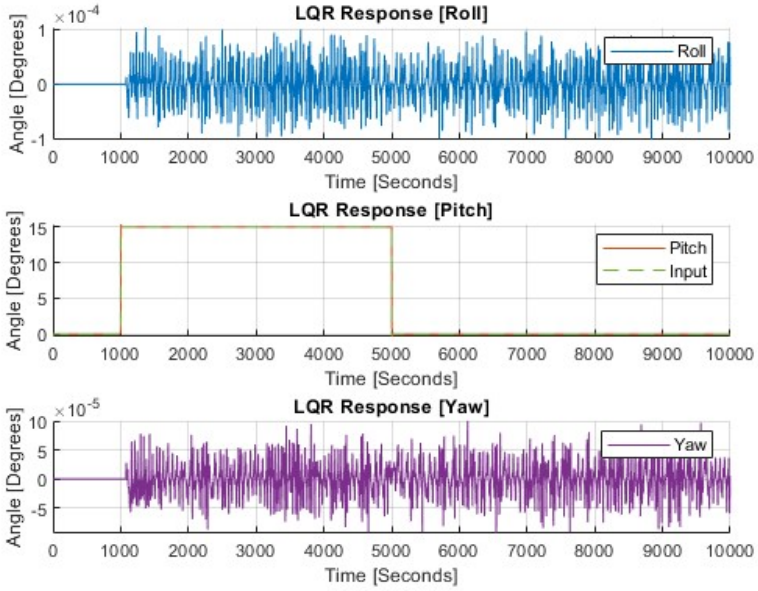

Figure 4.17: Spacecraft LQR-tracking pitch response

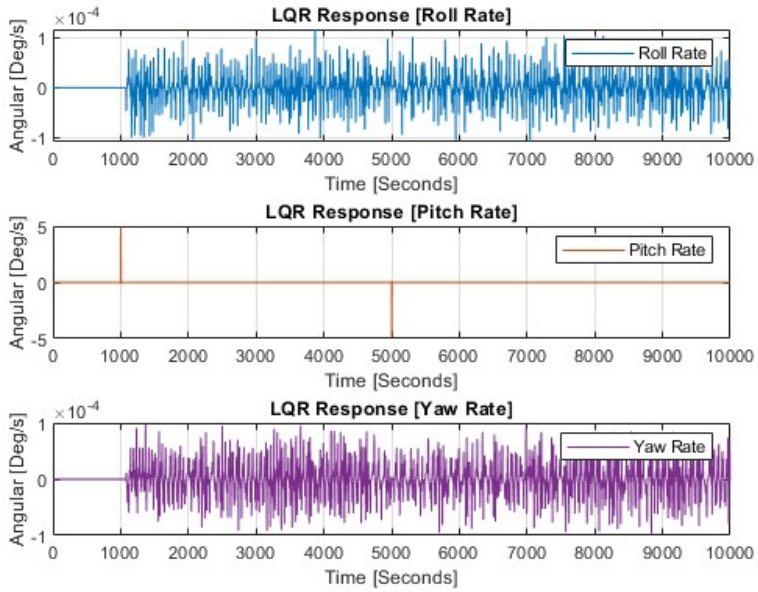

Figure 4.18: Spacecraft LQR-tracking pitch rate response

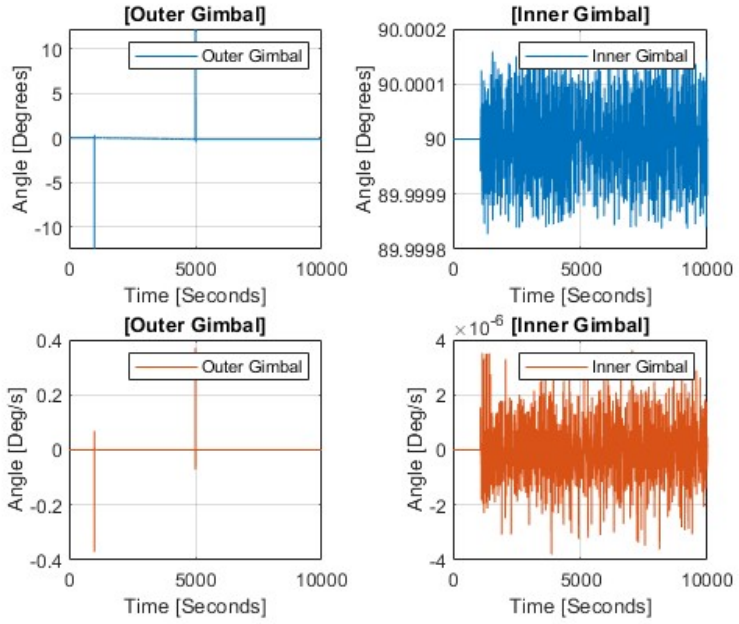

Figure 4.19: Spacecraft LQR-tracking pitch gimbal input

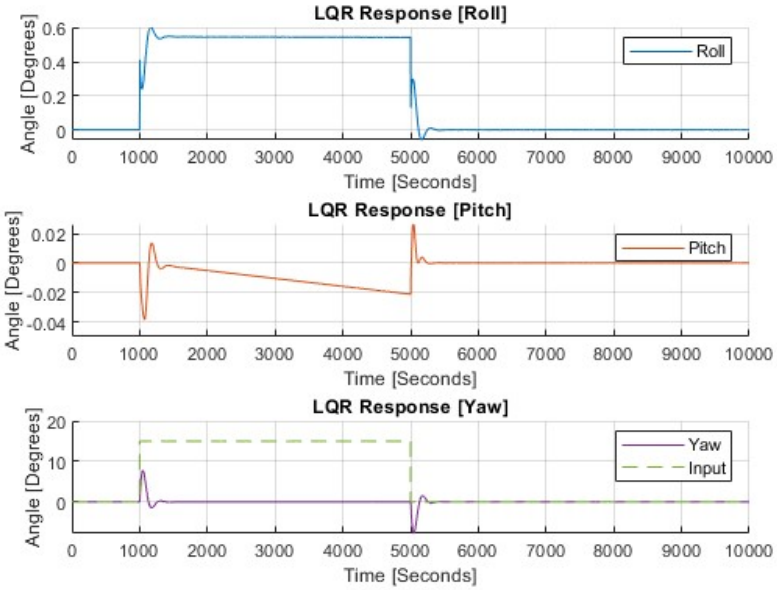

Figure 4.20: Spacecraft LQR-tracking yaw response

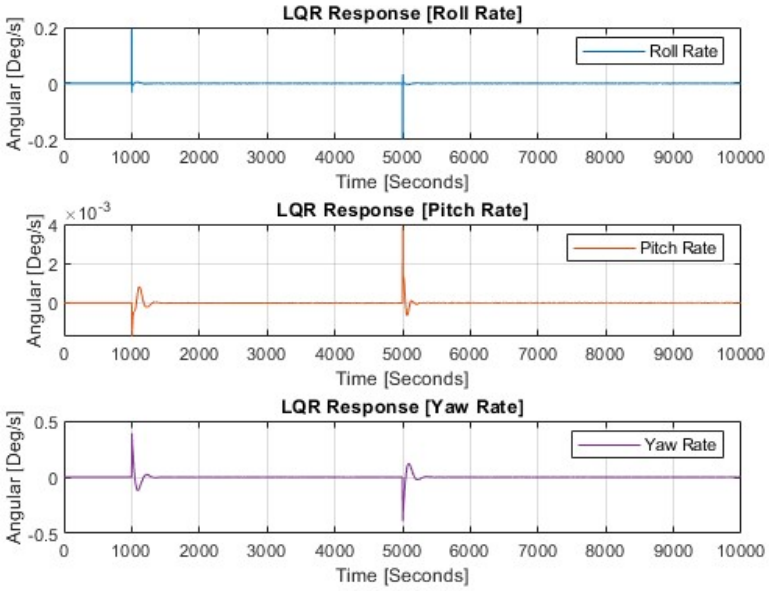

Figure 4.21: Spacecraft LQR-tracking yaw rate response

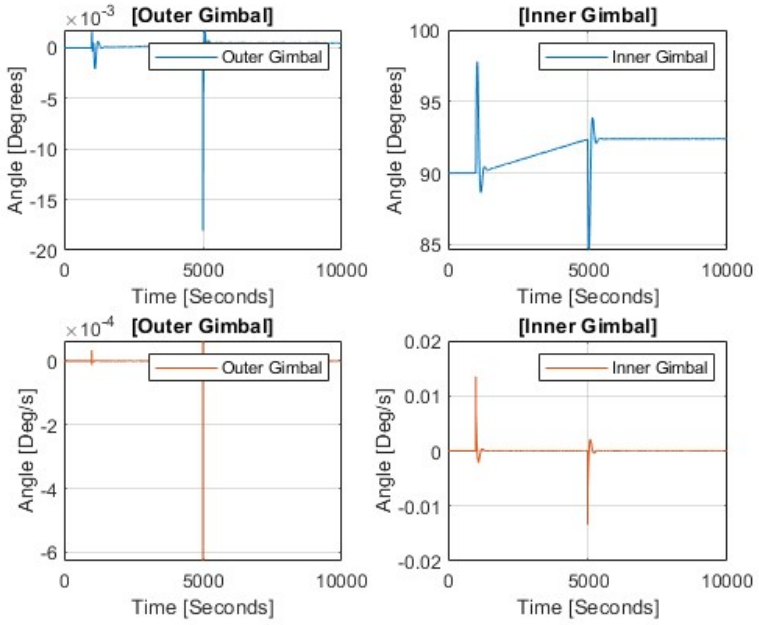

Figure 4.22: Spacecraft LQR-tracking yaw gimbal input

4.1.5 Closed-Loop Results for Testbed

Through the simulations provided for the testbed LQR system, the results of the system show that it is capable of turning to equilibrium conditions as seen in figures 4.3 and 4.4. Implementing the gain-scheduling controller was shown to schedule the corresponding gains to adjust the gimbal rates to necessary strengths dependent on the gimbal angles. Furthermore, adjusting the input signal for LQR-tracking was achieved in Figures 4.7 and 4.8. For this MISO LPV system, the desirable response for yaw was achievable by using the LPV model.

4.1.6 Closed-Loop Results for Spacecraft

Successful trackability for the roll and pitch of the spacecraft was simulated for the given impulse signal of 15 degrees as seen in Figures 4.14 and 4.17. The LQR-tracking can perform a responsive system that takes an average of 10 seconds to reach the desired angle. The gimbals rates of the servos were seen to be less than > 1 of a deg/s that being low considering the angular velocity threshold on the servos. However, holding the desirable angle requires a gimbal rate to counteract the effects of the gravitational torques induced on the spacecraft that potentially drive the gimbals of the DGCMG to their limits if held long enough. Hitting the saturation limits on the gimbal angle would cause the spacecraft to lose trackability and potentially its stability.

Using the DGCMG along with the LQR controller as a regulator to drive the orientation of the spacecraft to its equilibrium has been successfully simulated as shown in 4.10 and 4.11. The system is shown to have stability throughout its simulation for 30000 seconds, where it can passively maintain equilibrium without any additional actuation cost from DGCMG. Subjecting the LQR test to different initial conditions, it has been noted that the detumbling maneuver consumes a large portion of the available gimbal range limited by the DGCMG. This cost is influenced more when the initial angular rates are higher. Thus, this maneuver is costly to perform but offers quick stabilization for roll and pitch.

A method to desaturate the spacecraft is suggested to bring the gimbal servos of the DGCMG back to the initial orientation. Methods to desaturate the DGCMG require additional torques to counteract the momentum buildup on the spacecraft. Possible study methods can be done by using external torques such as solar radiation and gravity gradients [44]. The gravity gradient torque method to desaturate the DGCMG requires the spacecraft to be in specific conditions, such as being in LEO or performed when solar radiation pressure is neglectable [44]. More popular methods for desaturation are through the use of RCS to propel the CMG back to the needed orientation. A momentum dumping method would be required to autonomously drive the DGCMG back to initial conditions to perform desaturation.

One aspect that the controller wasn't able to achieve was the yaw angle. A desirable response for the yaw angle was not obtainable. As shown in Figure 4.20, the yaw had a minor reaction to the input signal. Despite expecting, Roll was seen to react to the input as it attempted to hold an orientation of approximately 0.6 degrees during tracking. Pitch had a minor response to the command.

5. Hardware and Software Development

5.1 Testbed Avionics

5.1.1 Communication Protocols

Communication protocols are standardized formats for how data is transmitted and exchanged among devices. It is essential to familiarize oneself with how these protocols work and check if the selected microcontroller supports the requirements. The testbed utilizes two types of communication protocols: USB and I2C.

I2C is a multi-slave serial bus commonly used on low-power consumption devices like sensors. It operates using two wires: a data-line SDA that transfers data and a clock-line SCL that synchronizes data transfers between the devices. With I2C, data from the IMU can be obtained for the ESP32.

USB is a high-speed wired communication protocol that supplies 5V power to any external device. The ESP32 uses a USB-C type connection intended for communications between the microcontroller and the device running the Integrated Development Environment software. The connection will be directly connected to the ESP32 and the computer and will be used for flashing and serial communication.

5.1.2 Power System

As seen in Figure 5.1, a generalized flowchart of the testbed has been laid out that describes the necessary voltage requirements for electronics. The testbed system will operate using a LIPO 11.1V battery as a DC power source, which has 2200mAh with a maximum discharge of 50C. 11.1V was selected such that the LIPO battery can be safely plugged into ESCs directly, as the D4215 Motor voltage requirement ranges from 7.4V to 14.8V. To supply components with lower voltages, buck converters are used to step down the 11.1V to 7.4V to supply power to the servos. Another buck converter is used to supply 11.1V to 3.3V for the ESP32 and IMU. Table 5.1 lists all necessary voltage requirements for the electrical components. For the testbed, understand that all components will be sharing the same ground pin. It is a requirement that the servos also share the same ground pins with the ESP32, as the breadboard test had issues operating the servos with multiple groundings.

Table 5.1: Hardware power requirement [10–15]

Hardware	Voltage Range
Adafruit ICM-20948	3.3V - 5V
DS3235 Servos	5V - 7.4V
ESP-WROOM-32 Dev Board	5V
30A ESC	7.4V - 14.8V
D4215 Motor	7.4V - 14.8V

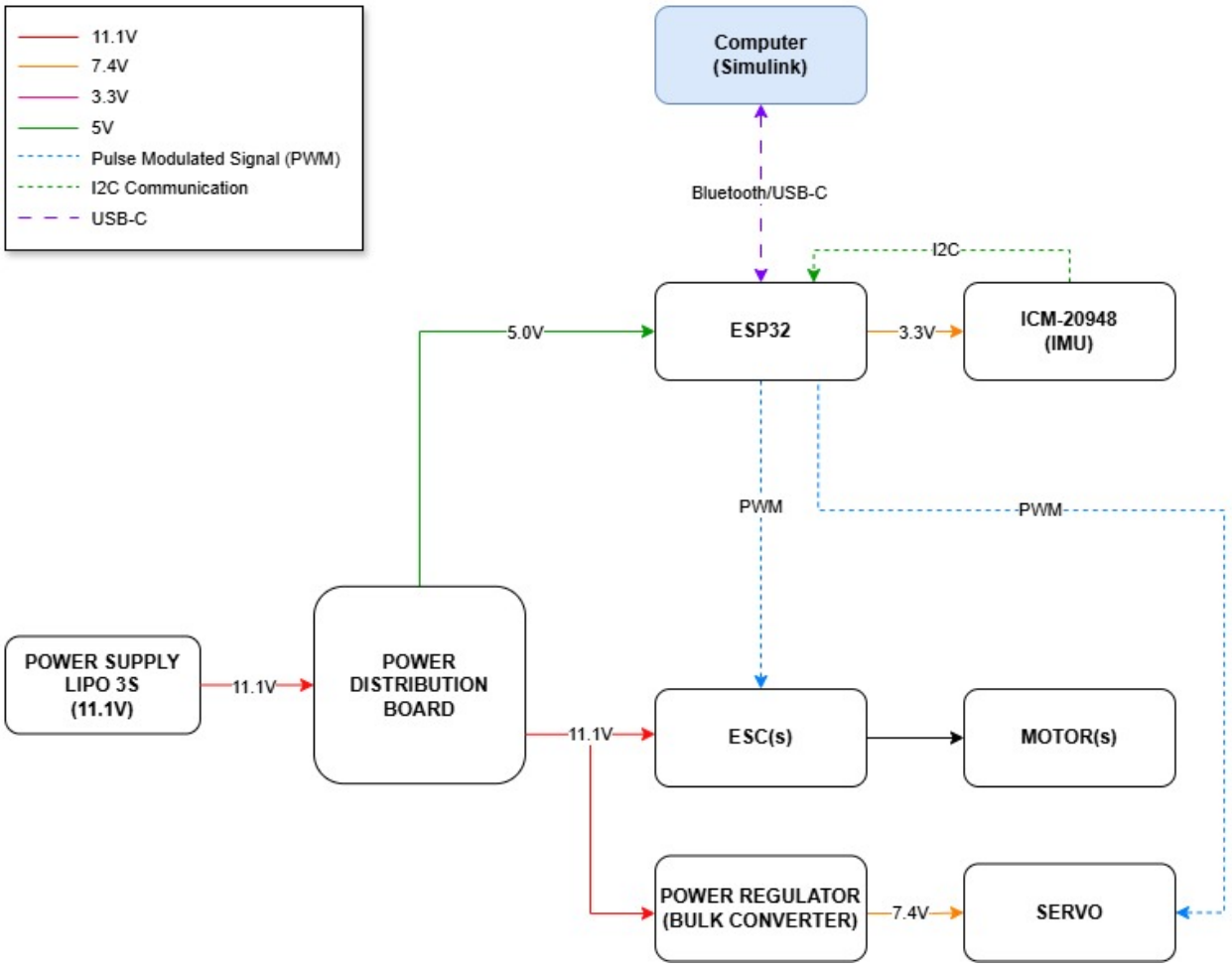

Figure 5.1: CMG control system architecture flowchart

Due to dimensional constraints of the testbed, a custom PCB was designed to ensure that the electrical components fit the provided dimensions for the payload of the electrical system. A custom PCB can help reduce required sizing, cable clutter, and ensure secure connections between components at its solder points and mounting with the base plate. It will be responsible for connecting the ESP32 with the ICM-20948, providing direct power for 5V and 7V, and KiCAD was used to design the PCB following the requirements of JLCPCB. Figure 5.2 references Figure 5.1 to layout the necessary configurations between each component.

After schematics, a dimension of the PCB is cutout is made for a 61cm x 61cm dimension with mountable M4 screw holes. Components and standard traces are routed to the necessary through-holes for each component at 1.532 mm. The 7V trace was designed with a 2.2 mm width on the copper back-layer to support higher currents drawn from the servo. The remaining spaces on the front and back copper layers were filled with a ground-plane layer.

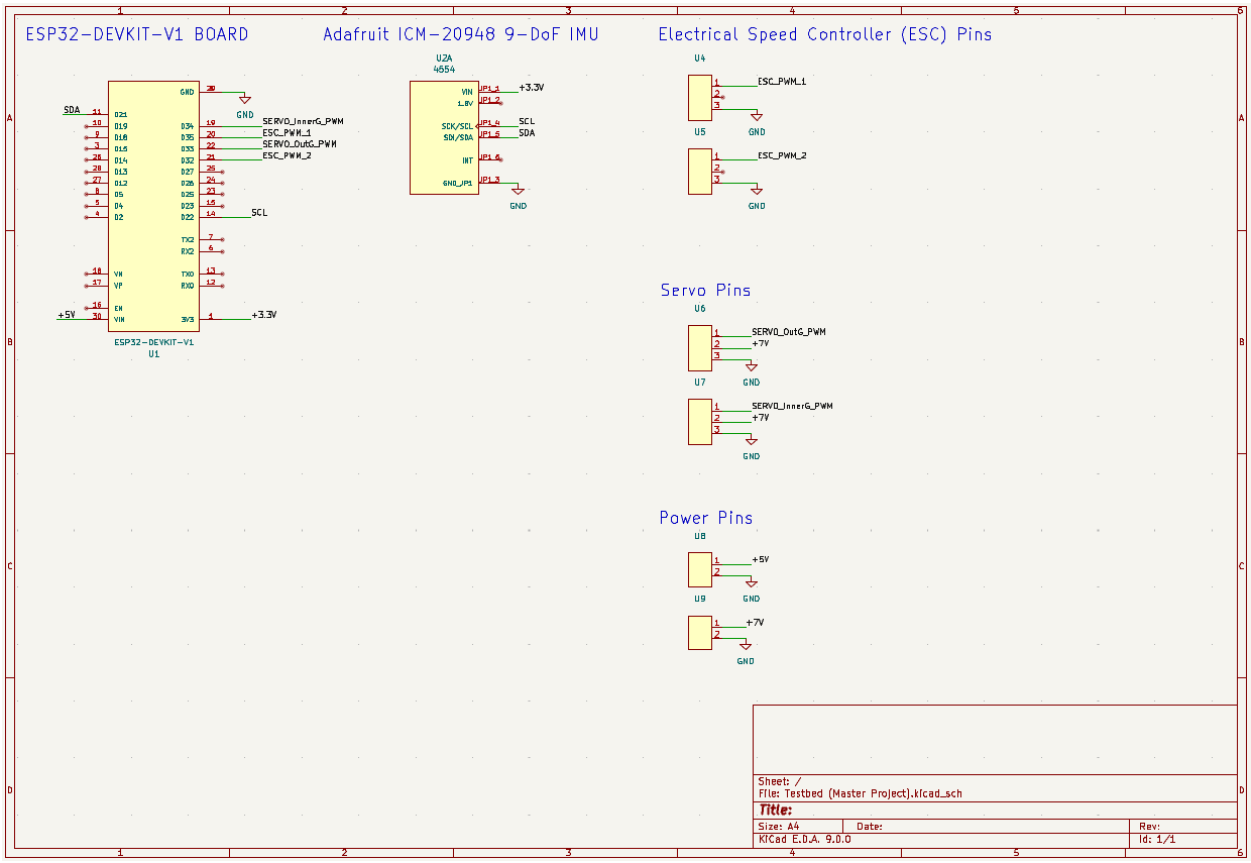

Figure 5.2: PCB schematic

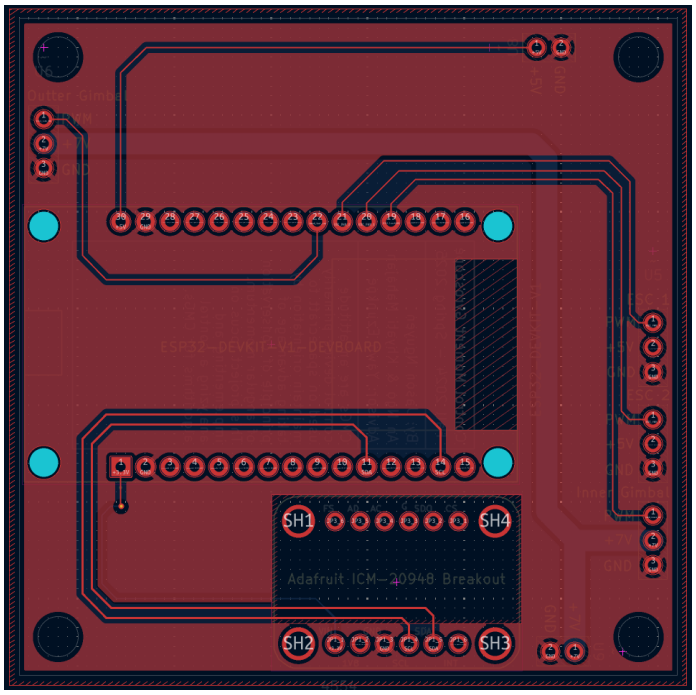

Figure 5.3: PCB front copper layer design

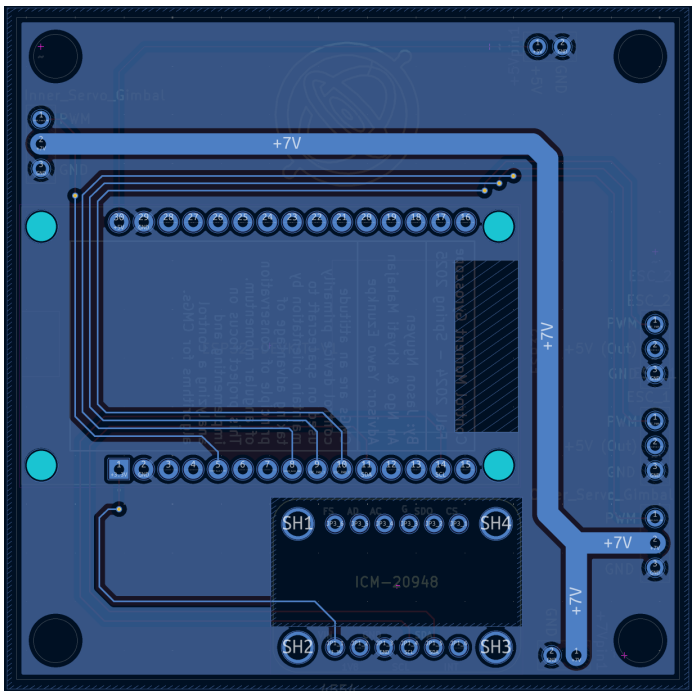

Figure 5.4: PCB back copper layer design

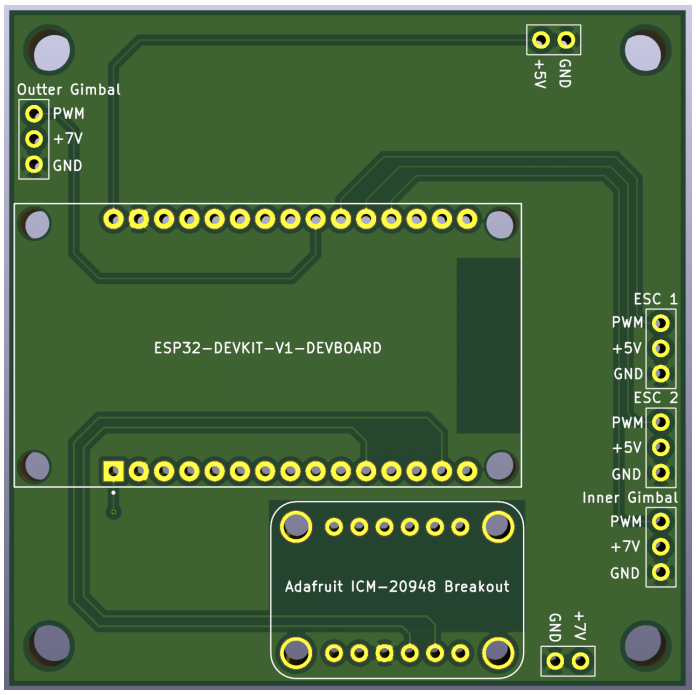

Figure 5.5: PCB 3D front rendered model

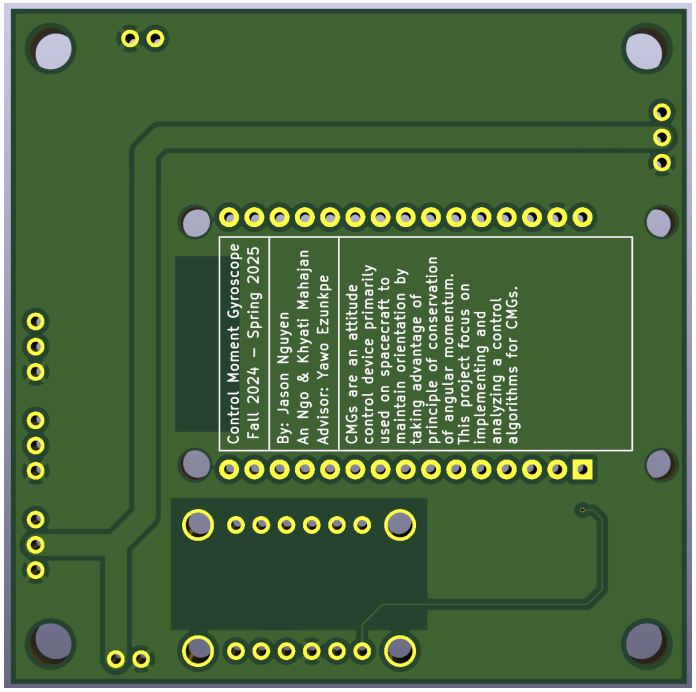

Figure 5.6: PCB 3D back rendered model

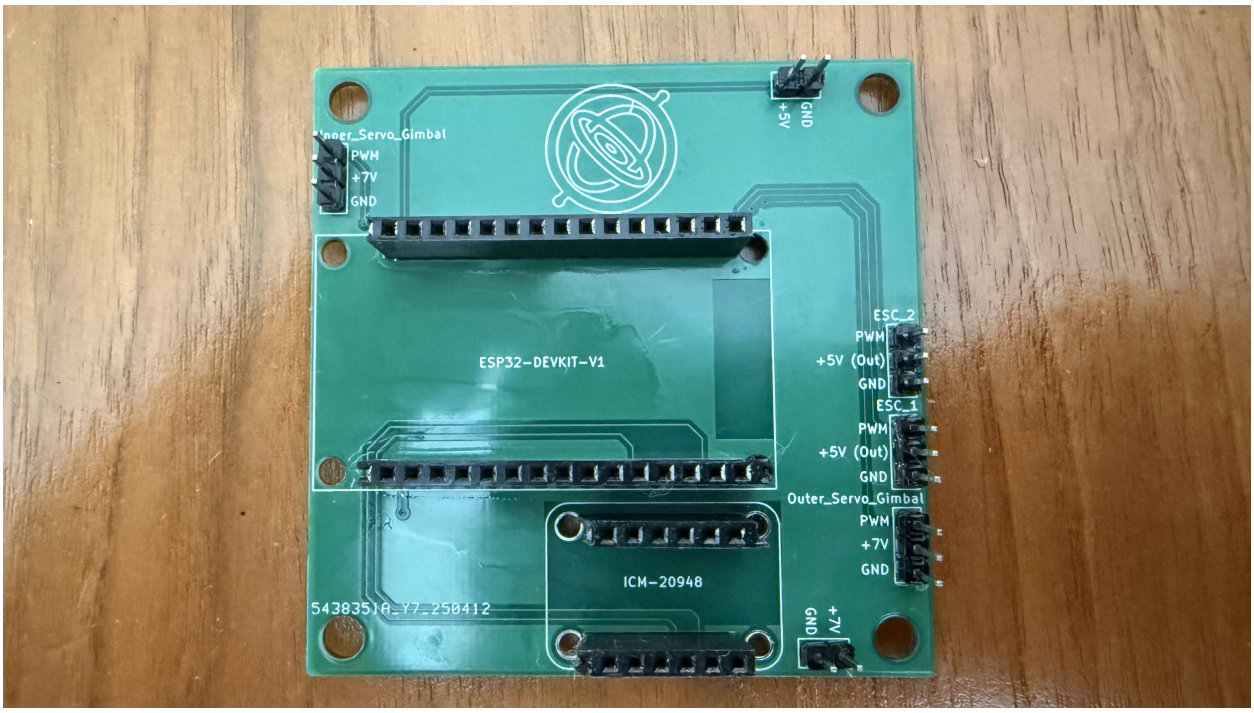

Figure 5.7: Manufactured PCB

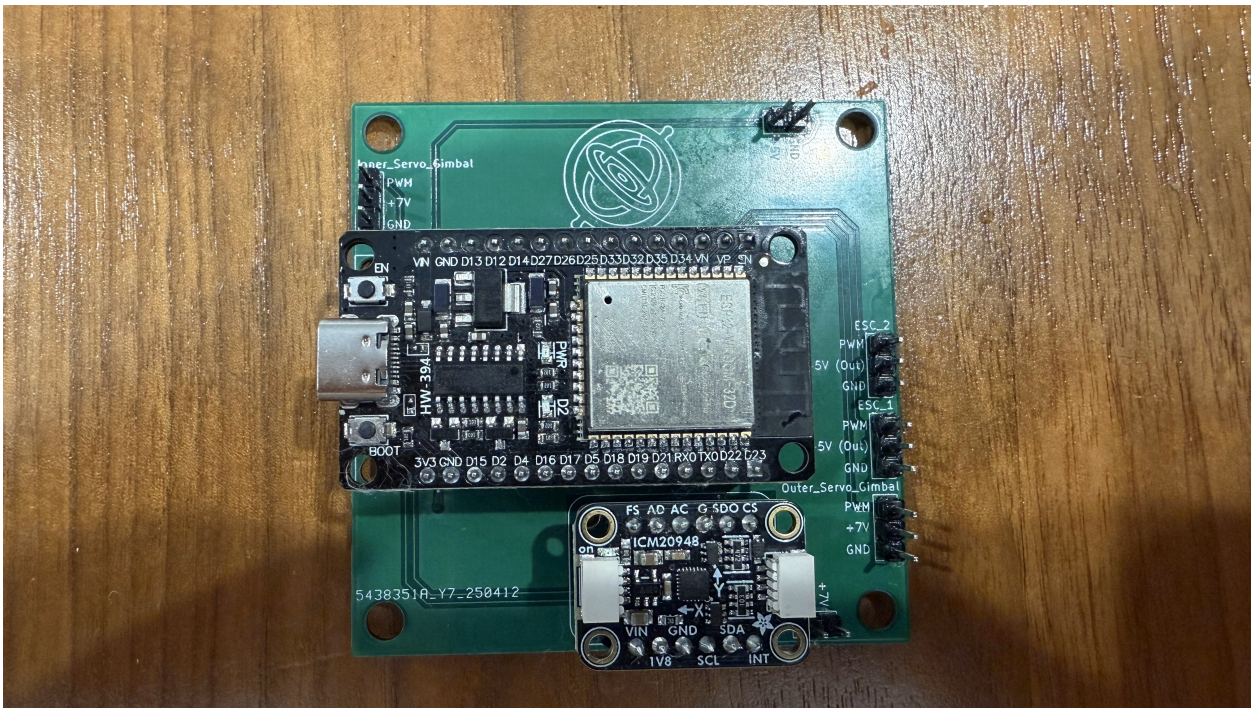

Figure 5.8: Manufactured PCB with breakout boards

5.2 Programming

To design the testbed program, a flowchart is used to help portray the sequence of operations and system interaction as seen in Figure 5.9. Each operation carries out a specified task that is either responsible for user inputs, initializing electrical components, or computing control inputs. The sequence is designed to align the position of the servo horizontally with the testbed so that the effects of angular momentum about the z-axis for the testbed would be negligible when the motors initialize. After the motor, calibration for the IMU is initialized. A feedback loop is then designed to provide the IMU yaw data to the Simulink model to calculate the desirable control inputs for the servos. The majority of the codes presented in this section are provided in the GitHub repository [45].

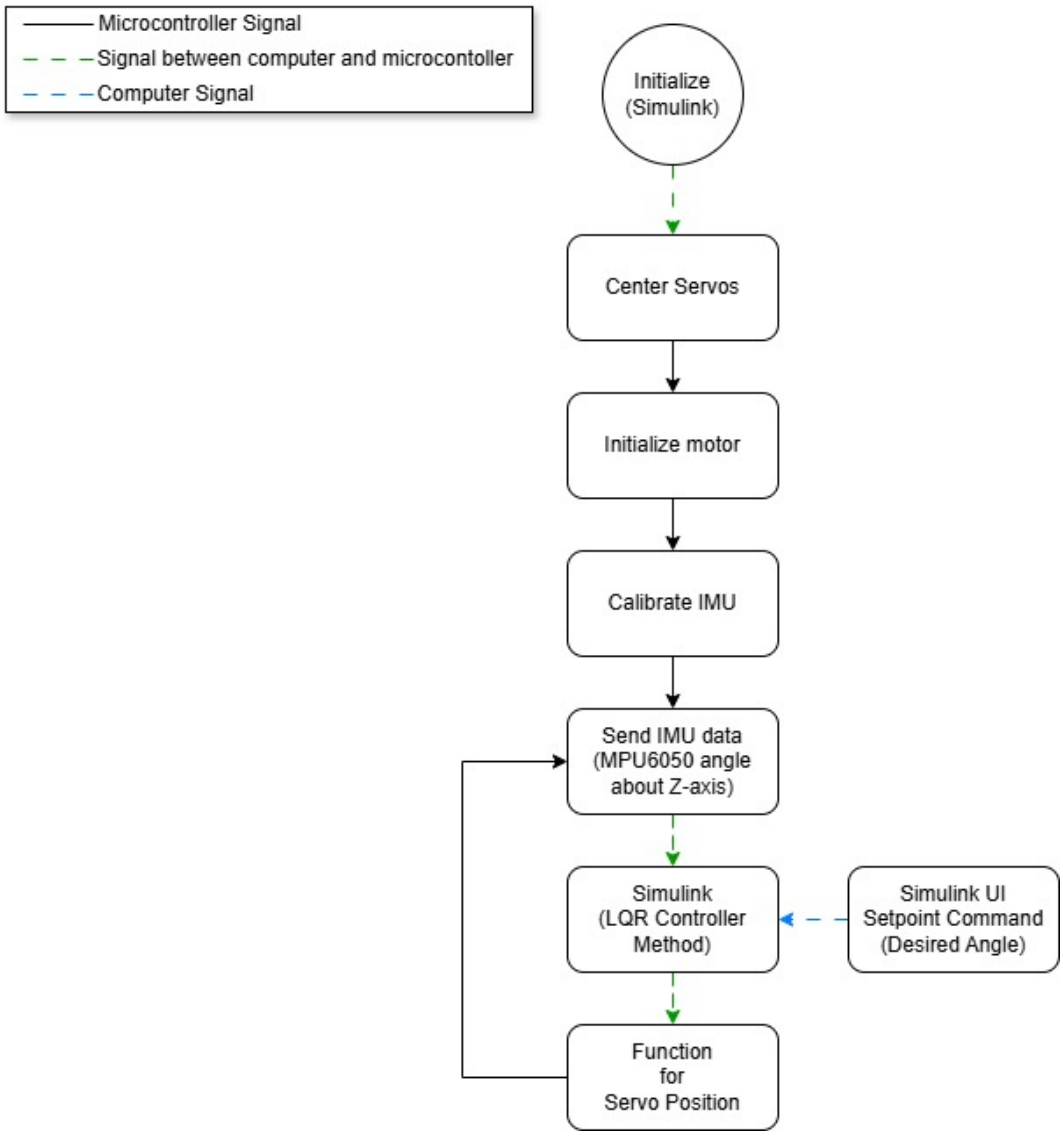

Figure 5.9: Sequence of operation flowchart

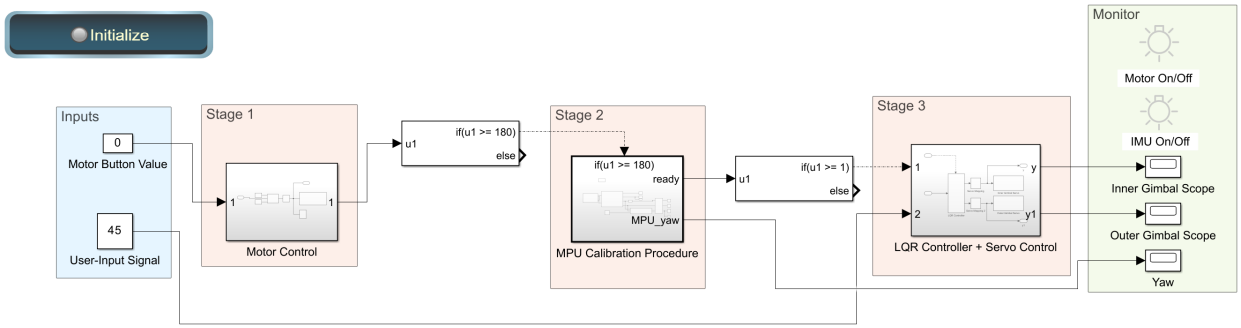

Figure 5.10: Simulink model for testbed hardware

5.2.1 Electronic Speed Control & Motor

An electronic speed controller (ESC) and motor are used to control the angular momentum of the two flywheels by managing the desired angular velocity. An electronic speed controller provides power to the brushless direct current motor (BLDC) through three-phase PWM signals. A back-electrical magnetic field (EMF) is generated by the motor as it spins, which can be sensed by the ESC to measure the position of the spinning motor without using a hall sensor or encoder. This back-EMF is then used to provide feedback to the ESC to control the desired speed. To set the desired speed on an ESC, an 8-bit PWM signal is received via the microcontroller that ranges from 0 to 255 to correspond to the speed.

To get the ESC working with MATLAB Simulink, a servo library block bounded input signal is designed to initially arm the motor when the input is equal to 53 degrees and output the maximum speed when 180 degrees is set. This boot-up sequence can be seen in Figure 5.11. This adjustment was required to consolidate the ESC operating PWM frequency of 50Hz and required an initial frequency to arm the motor properly. For the motor and ESC models, a D4215 BLDC was selected to pair with a 30A ESC controller.

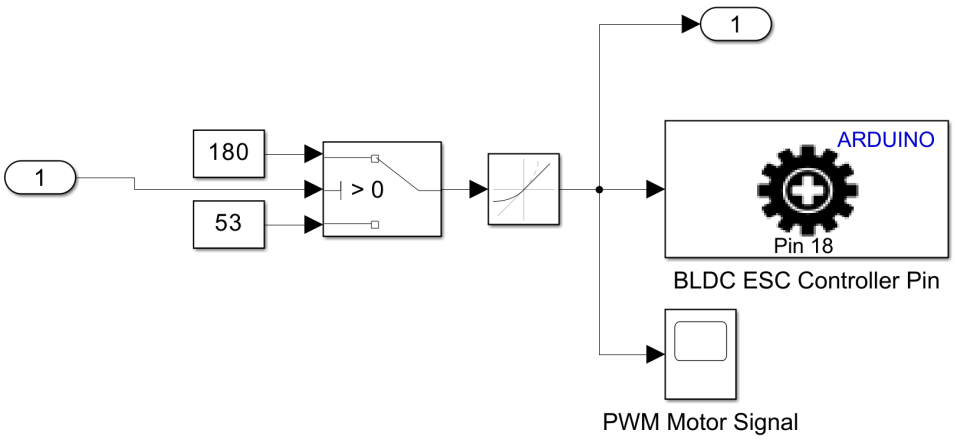

Figure 5.11: Simulink ESC and motor model

5.2.2 Servo

Two servos are used to control the angular position of the gimbals. A servo operates using a motor and a potentiometer as part of a closed-loop system to drive the servo shaft to the desired angular position. Two DS3235 position servos have been selected for the prototype with a 270-degree control angle and a high peak torque of 29 kg/cm. Though the gimbal limits are set to 180 degrees, a 270-degree servo can adjust for offsets that off-center the gimbals, caused by the servo hooks and mounting of the gimbal. The specifications of DS3235 are listed in Table 5.2.

Table 5.2: DS3235 servo [10]

Parameter	Value
Operating Voltage Range:	5V - 7.4V
Operating speed:	500 deg/s
Stall torque:	29 kg/cm
Dimensions:	40 x 20 x 38.5 mm

Programming the servo with a range of 270 degrees in Simulink will require adjustments to the input signals. The servo block seen in Figure 5.12 is programmed to send a PWM signal ranging from 2000 Hz to 400 Hz to command a 180 degree servo. Input signals for the servo block must range from 0 to 180 degrees. To account for a servo with a 270 degree range, a mapping function is used to translate the 0 to 270 degree to a 0 to 180 degree input signal. An offset of 135 degrees was required such that the servo can accept signals from -90 to 90 input signal received from the LQR controller, and have the servo initially start at its center point.

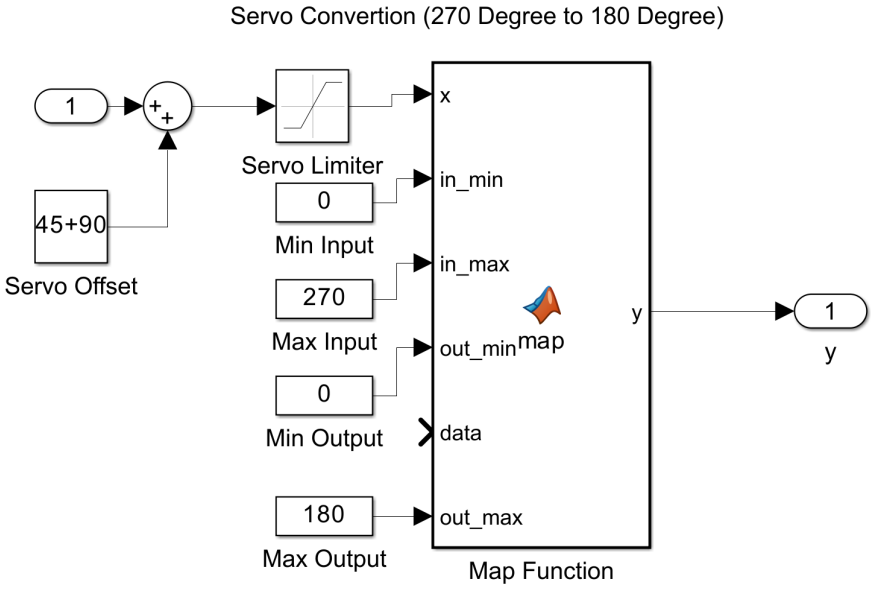

Figure 5.12: Simulink model for servo mapping

5.2.3 Inertial Measuring Unit

An inertial measuring unit (IMU) is a sensor system that utilizes an accelerometer, gyroscope, and magnetometer to measure the orientation at which it is placed. It is popular in many maneuvering applications found in robotics and aerospace. 3-axis is given to each sensor in its naming convention, thus it is usually popular to see 6-axis or 9-axis motion tracking. An IMU provides a versatile method to determine motion in its inertial surroundings without the need for a fixed reference point (apart from Earth's magnetic field). IMUs provide fast and continuous measurements for rapid movements, are relatively small, and are generally known to be suitable for short-term precision tracking. Under perfect ideal conditions, an IMU would serve as a perfect application to precisely measure translational position and rotation. However, in reality, IMUs are subject to various conditions, including noise, vibrations, temperature, and electromagnetic interference. These factors induce drift and bias that hinder the accuracy of the measurements.

To combat these issues, calibration methods are required to obtain reasonable results for position and orientation. Sensor fusion can be used to combine multiple data sources from the sensor to help improve the accuracy, consistency, and reliability of the system. For example, GPS and IMU data are combined to determine the localization and positioning of the system. For the IMU, a discussion on calibrating the gyroscope, accelerometer, and magnetometer will be provided. Performing the proper procedures will allow for a calibrated IMU that can accurately measure the orientation and provide readings for the attitude angles for the testbed.

5.2.3.1 Gyroscope

A gyroscope sensor measures the angular rate in rad/s for pitch, roll, and yaw. Obtaining the orientation of the IMU requires integration of the angular rates and is generally good for high frequencies. Gyroscopes are known to have a bias in the data that is present when the IMU is stationary. Corrections are required for the bias to be subtracted. In addition, integrating the angular rates is required to obtain the angles from the gyroscope. However, applying this integration method to hardware will result in a gradual bias on the angle known as drift.

To deal with the bias on a gyroscope, a dataset of the raw gyroscope data is collected while the IMU is stationary. This dataset is averaged out to obtain the $\omega_{stat,gyro}$ and is subtracted from the raw measured gyroscope seen in Equation 5.1 to obtain the calibrated angular rate for the gyroscope. The calibrated angular rate can be integrated to approximate attitude angles.

$$\omega_{cal,gyro} = \omega_{meas,gyro} - \omega_{stat,gyro} \tag{5.1}$$

5.2.3.2 Accelerometer

Accelerometers are sensors designed to measure influences in linear acceleration. Sources can range from direct motion, vibrations, and acceleration from gravity. Accelerometers are suitable for low-frequency applications compared to a gyroscope. To obtain the orientation of the IMU, the acceleration vectors from gravity are used to solve for pitch and roll. Equations 5.2 are used to determine the angles between the gravitational vector with respect to the body frame of the IMU. The heading angle (yaw) is unobtainable as the gravity vector will always be oriented along the z-axis of the inertial frame.

Calibrating the accelerometer is similar to that of the gyroscope, but it needs to ensure that the gravitational vector is present when measured. To perform this task, a dataset in terms in units of m/s^2 is collected onto a .txt file and imported onto the software "Magneto" where calculations are performed to obtain the combined offset θ_{bias} and scaling factor A_{acc}^{-1} seen in 5.4 and 5.5. The scaling factor is a symmetric matrix that describes ratios that quantify the error between the actual sensor output and the measured input change. Scaling for the accelerometer will normalize the vector and adjust for slight distortions in the data. Equation 5.3 is the relationship for correcting accelerometer data by multiplying the scaling factor by the output signal, subtracted by the bias [46]. Inputting the calibrated data into Equation 5.2 will provide an accurate measurement for orientation.

$$\begin{aligned} \phi_{acc} &= tan^{-1}\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \\ \theta_{acc} &= tan^{-1}\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \end{aligned} \tag{5.2}$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_{cal,acc} = A_{acc}^{-1} \cdot \left(\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_{acc} - \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_{bias} \right) \tag{5.3}$$

$$A_{acc}^{-1} = \begin{bmatrix} 1.003628 & 0.064442 & -0.008783 \\ 0.064442 & 1.002207 & -0.000245 \\ -0.008783 & -0.000245 & 0.994195 \end{bmatrix} \tag{5.4}$$

$$B_{acc} = \begin{bmatrix} 0.016162 & 0.027335 & 0.093360 \end{bmatrix} \tag{5.5}$$

To provide a visual of the bias and distortions, a scatterplot can be used to plot the raw normalized accelerometer data with the calibrated accelerometer data. A visualization of an ellipsoid fitting of both the uncalibrated and calibrated data in Figure 5.13 that is meant to show the difference between the two sets. Observing the difference between the datasets is noticeably small for the accelerometer, but a noticeable offset can be identified. Calibrated data in the figure shows a visualization of the adjustment made to correct the data back to the origin.

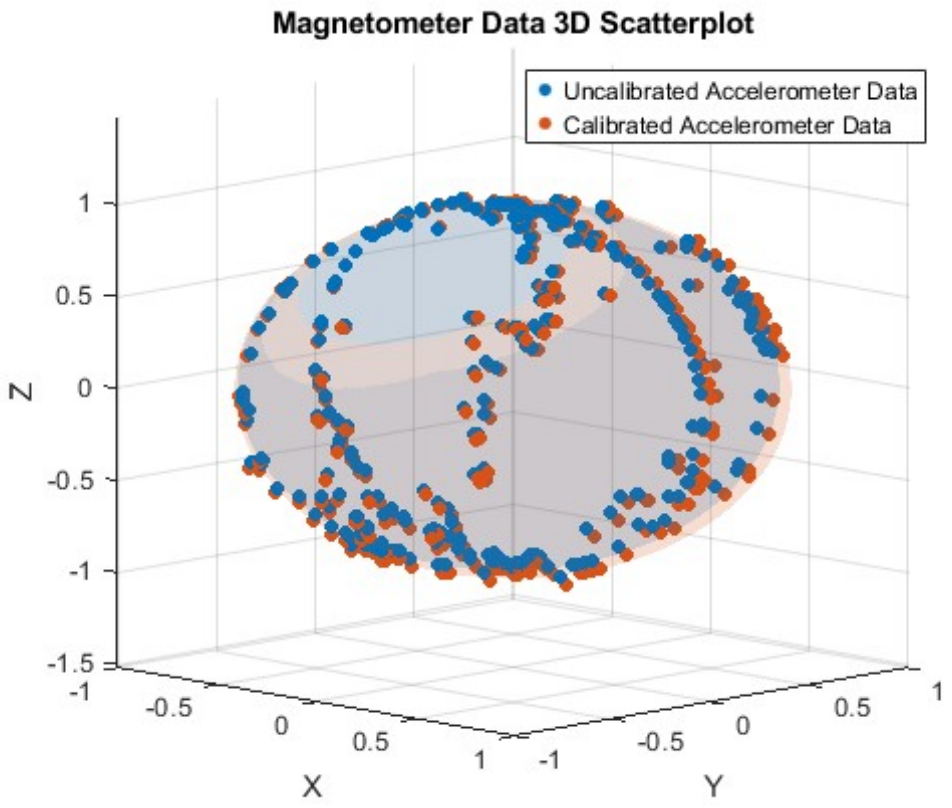

Figure 5.13: Scatterplot of raw vs. calibrated normalized accelerometer data

5.2.3.3 Magnetometers

Magnetometers are used to measure the strength and direction of a magnetic field. In many applications for robotics and aerospace, magnetometers are used primarily to determine the heading angle (yaw) of the system by detecting the Earth's magnetic field, similar to how a compass works. Calibrating the magnetometer data shares the same procedures as the accelerometer. Magneto is used to calibrate the bias and scaling factor using the raw magnetometer data with a scale factor of 1. Note that the scaling factor takes into account the hard and soft iron distortion caused by interferences from other magnetic sources and ferromagnetic materials. Scaling factor and bias offset are provided in equations 5.8 and 5.9, respectively. Once the data has been calibrated, a heading angle can be calculated by using Equation 5.6 in units of degrees. Note that Equation 5.6 will have quadrant issues as tan^{-1} is bounded between $-\pi/2$ and $\pi/2$, resulting in incorrect readings if the IMU passes these bounds. Thus, the two-argument arc tangent function atan2() is applied for a four-quadrant inverse, allowing for corrections in computing the correct angle seen in Equation 5.7.

$$\theta_z = tan^{-1}(\frac{m_y}{m_x}) \tag{5.6}$$

$$\theta_z = atan2(\frac{m_y}{m_x}) \tag{5.7}$$

$$A_{mag}^{-1} = \begin{bmatrix} 0.101925 & -0.000025 & -0.000179 \\ -0.000025 & 0.102289 & -0.000287 \\ -0.000179 & -0.000287 & 0.101810 \end{bmatrix} \tag{5.8}$$

$$B_{mag} = \begin{bmatrix} -22.141157 & -6.868126 & -30.821610 \end{bmatrix} \tag{5.9}$$

Scatterplot for the magnetometer can be seen in Figure 5.14, where normalized uncalibrated and calibrated data are shown. The ellipsoid is much defined compared to the accelerometer scatterplot. Offsets and distortions are much more pronounced, shown by uncalibrated data failing to normalize to 1 and offsetting away from the origin. By contrast, the calibrated data shows the correction in this offset and bias by re-centering the data back to the origin and scaling the data back to 1.

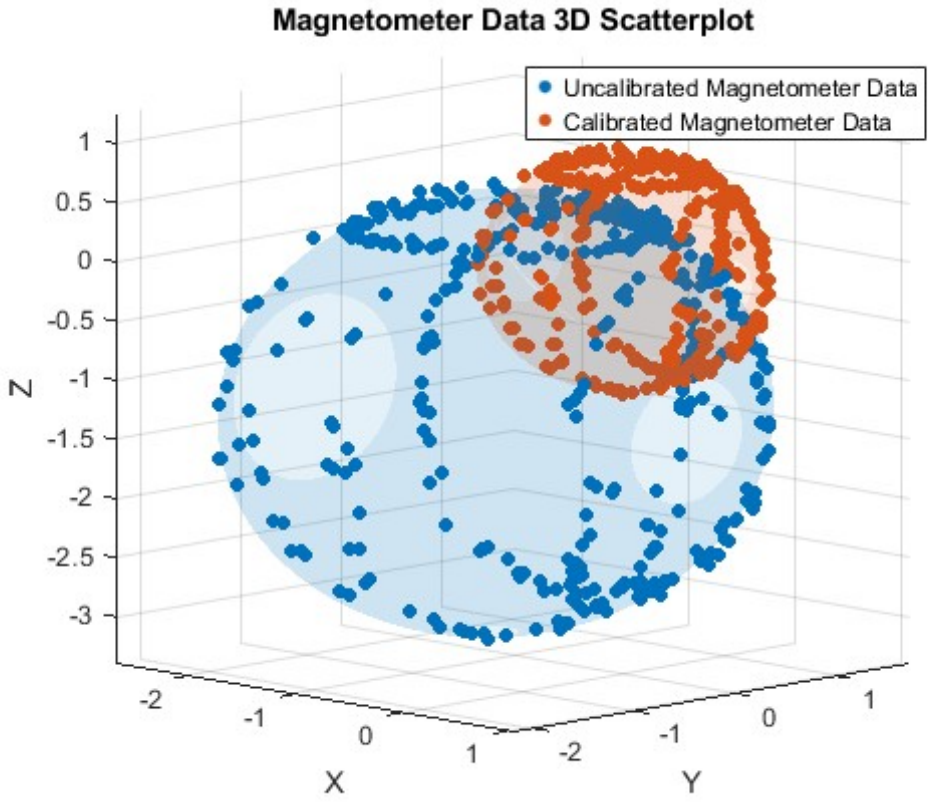

Figure 5.14: Scatterplot of raw vs. calibrated normalized magnetometer data

5.2.3.4 Complementary Filters

Effectively utilizing the sensor readings requires filtering to mitigate the reliance on using a single sensor to determine orientation. A complementary filter essentially combines readings from two sensors, thus leveraging the strengths and compensating for the weaknesses of each sensor. It is commonly known for combining low and high frequency data and is good for basic applications such as balancing a pendulum and balancing reaction wheels. To apply a filter with the IMU, gyroscope and accelerometer data are used to determine the pitch and roll angles. Combining the gyroscope and magnetometer readings is used to determine the yaw angle.

A straightforward implementation of the complementary filter is shown in Equation 5.10 [47]. α is the weighting factor that shifts the influences of each sensor and is typically tuned between 0.95 to 0.98 such that gyroscope readings will have significant influence and accelerometer readings will provide long-term corrections from bias. Δt is the timestamp between the previous and current readings.

$$\psi_n = \alpha \cdot (\psi_{n-1} + \dot{\psi}_{cal,gyro}\Delta t) + (1-\alpha) \cdot \psi_{cal,acc} \tag{5.10}$$

Coding the complementary filter in MATLAB Simulink was done using a MATLAB function block, and a test was conducted to evaluate the effectiveness and robustness of the ICM-20948. The results seen in Figure 5.16 test the orientation for pitch, roll, and yaw. Along each axis, the ICM-20948 was oriented 90 degrees and -90 degrees from its stationary orientation to determine if the data being read is accurate. As the ICM-20948 is a 9-axis measuring unit, readings for each pitch, roll, and yaw should be feasible. Testing the hardware component, observation can be seen on how the complementary filter can quickly track the high response of the gyroscope while slowly converging to the accelerometer data to correct bias. Furthermore, pitch and roll were shown to track 90 degrees accurately during the test. If yaw is tested independently, the 90-degree test successfully combines the gyroscope and magnetometer data through the complementary filter. If rotating the pitch and roll, a noticeable change in the heading angle of the magnetometer is picked up, which inaccurately influences the results. For Figure 5.16, testing yaw occurs between 35 and 50 seconds during the simulation.

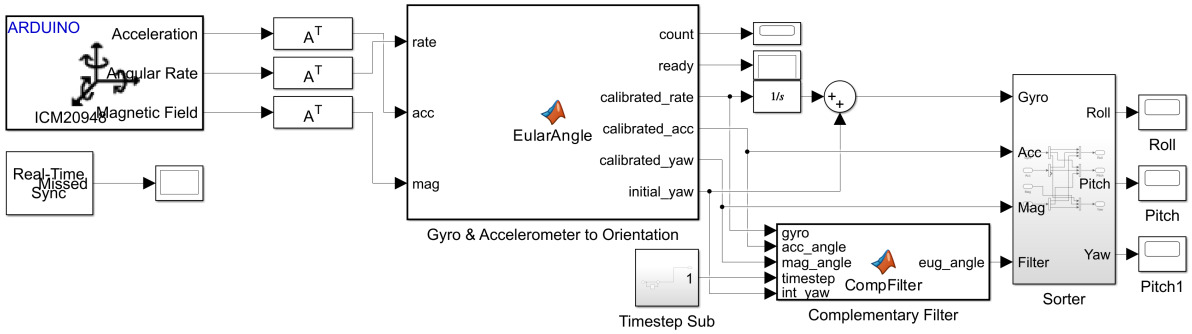

Figure 5.15: Simulink structure for calibrating ICM-20948 with complementary filter

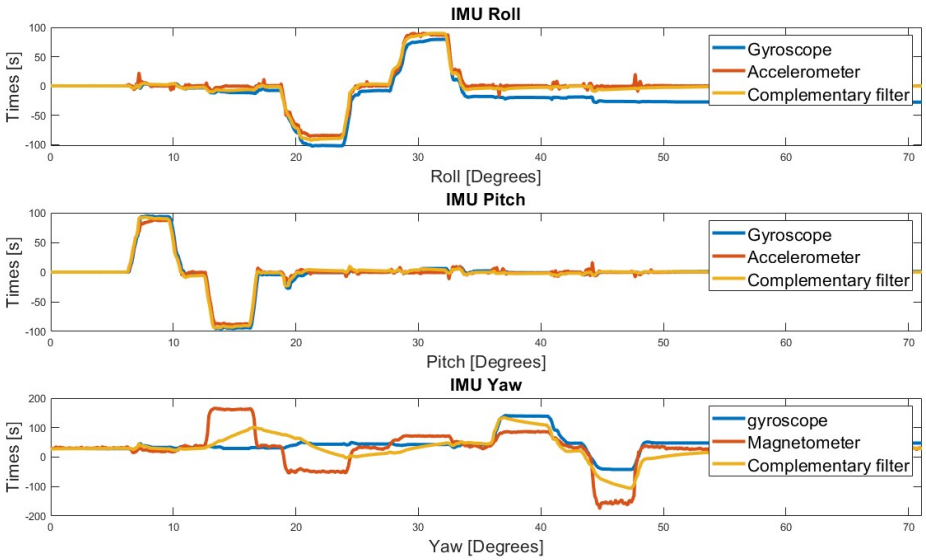

Figure 5.16: ICM-20948 calibrate data for roll, pitch, and yaw

For the model seen in Figure 5.15, noticeable counts in maximum missed ticks increase gradually and eventually cause the model to shut down when running in I/O mode. To solve this issue, a reduced Simulink model was specifically designed to measure the yaw angle for the testbed and optimize the performance as seen in 5.17.

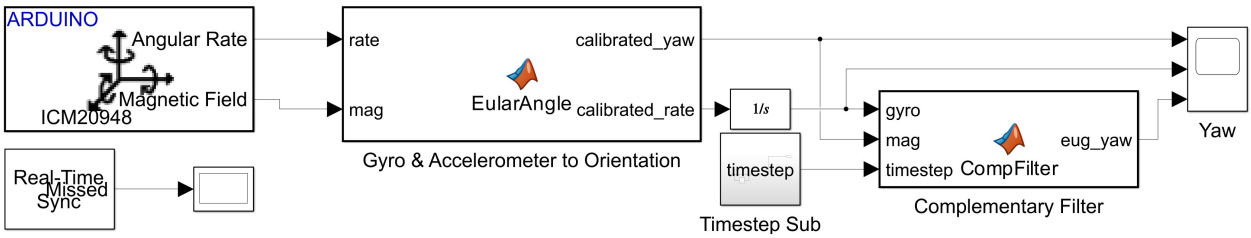

Figure 5.17: ICM-20948 reduced Simulink model for yaw/heading angle

5.2.4 Control Algorithm

Modifications for the control algorithm were made using the LPV model seen in 5.18 for the testbed. These modifications included direct gimbal angles inputs to the gimbal servos and feedback of the filtered yaw readings from the IMU. As the DS3235 servo can be controlled only by specifying the position of the servo, it was necessary to input the position of the gimbals rather than the gimbal angular rates. A continuous servo could suffice in this requirement, but would trade off with the high torque provided by the DS3235 servo. IMU would be necessary to provide physical readings from the actual testbed, such that the control algorithm can compute the control inputs required to actuate the device about its yaw-axis.

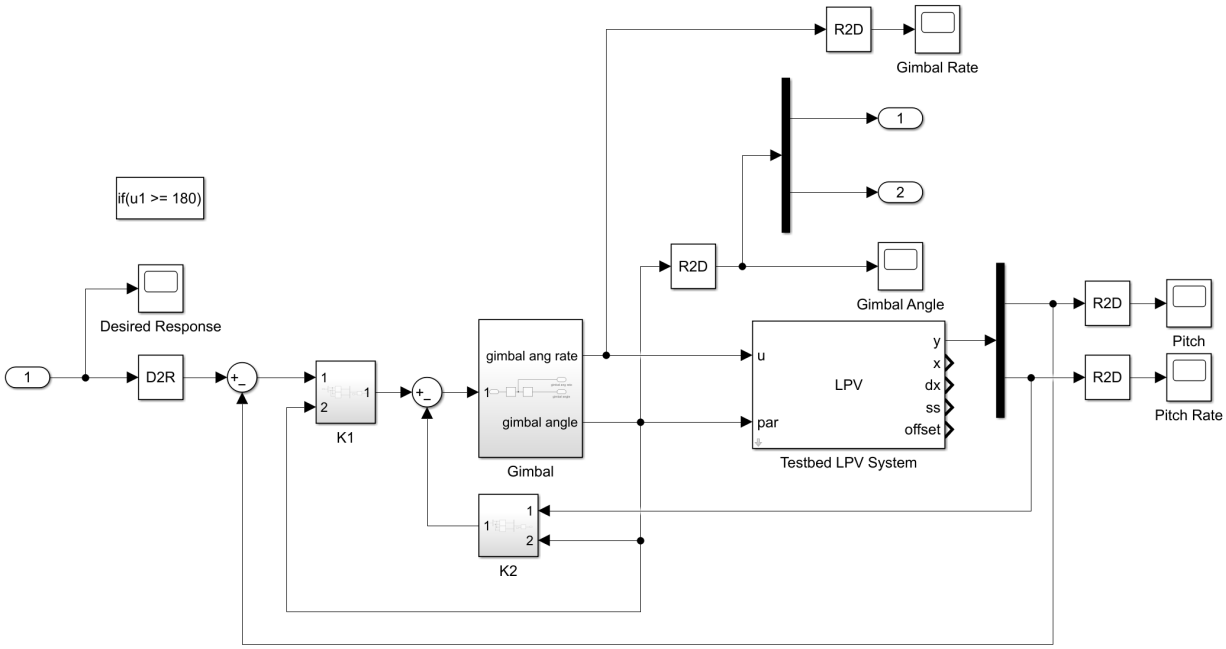

Figure 5.18: LQR-tracking for testbed prototype

6. CAD and Assembly

Designing the physical components for the testbed concentrated on the constraints of the air bearing dimensions. As the device is intended to essentially hover and rotate above the air bearing, the lateral dimensions of the device should not exceed the radius of the air bearing. The air bearing provided to this project is a 200mm precision flat round air bearing by New Way Air Bearing. These air bearings are commonly used for designed applications in linear or rotational motions. The specifications for the 200mm air bearing that were considered are provided below in Table 6.1.

Table 6.1: New Way air bearing constraints [12]

Parameter	Value
Operating Voltage Range:	5V - 7.4V
Bearing Face Surface Size	197 mm
Ideal Load	7770 N
Input Pressure	60 psi
Max Pressure	100 psi

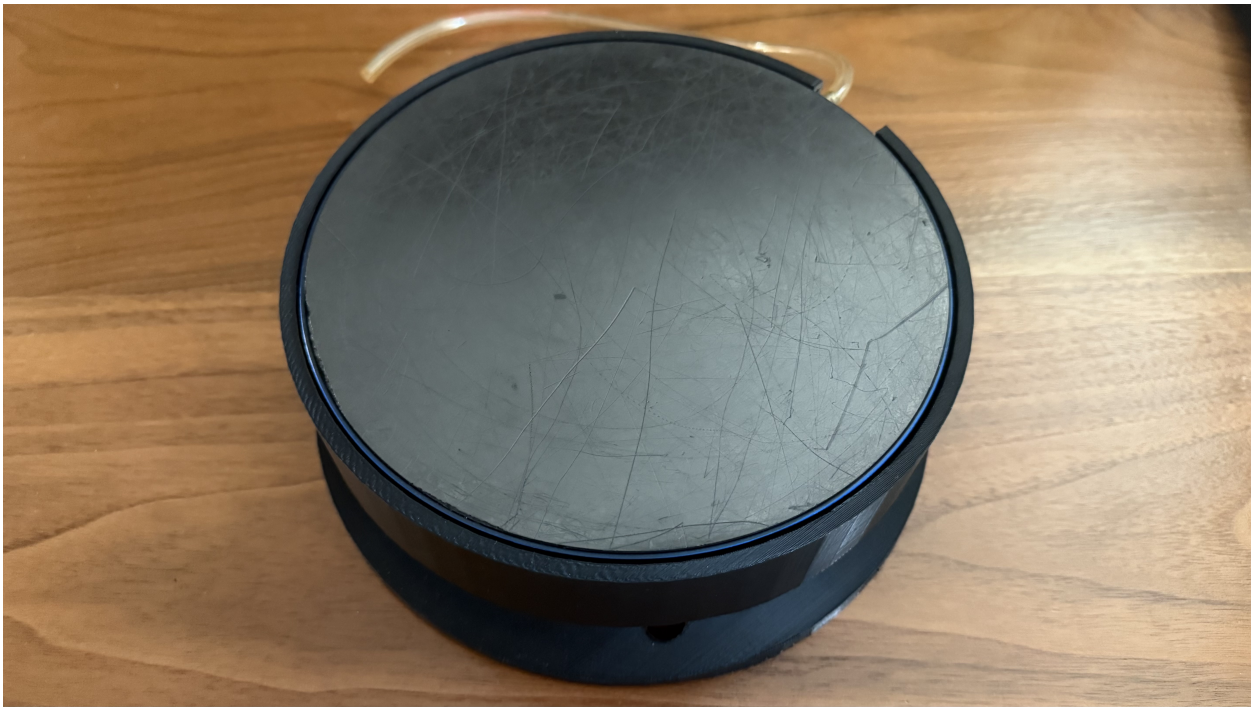

Figure 6.1: Air bearing & 3D printed shell

As the base plate is planned to be 3D printed, it is expected that the surface of the 3D printed base plate will not be smooth enough to work properly with the air bearing. For the testbed to function properly, a smooth flat surface is generally required. This flat surface is adhesive mounted to the testbed base plate and will sit on top of the air bearing porous material seen in Figure 6.1. A 197 mm cylindrical sheet of stainless steel in Figure 6.2 will adequately serve for this purpose.

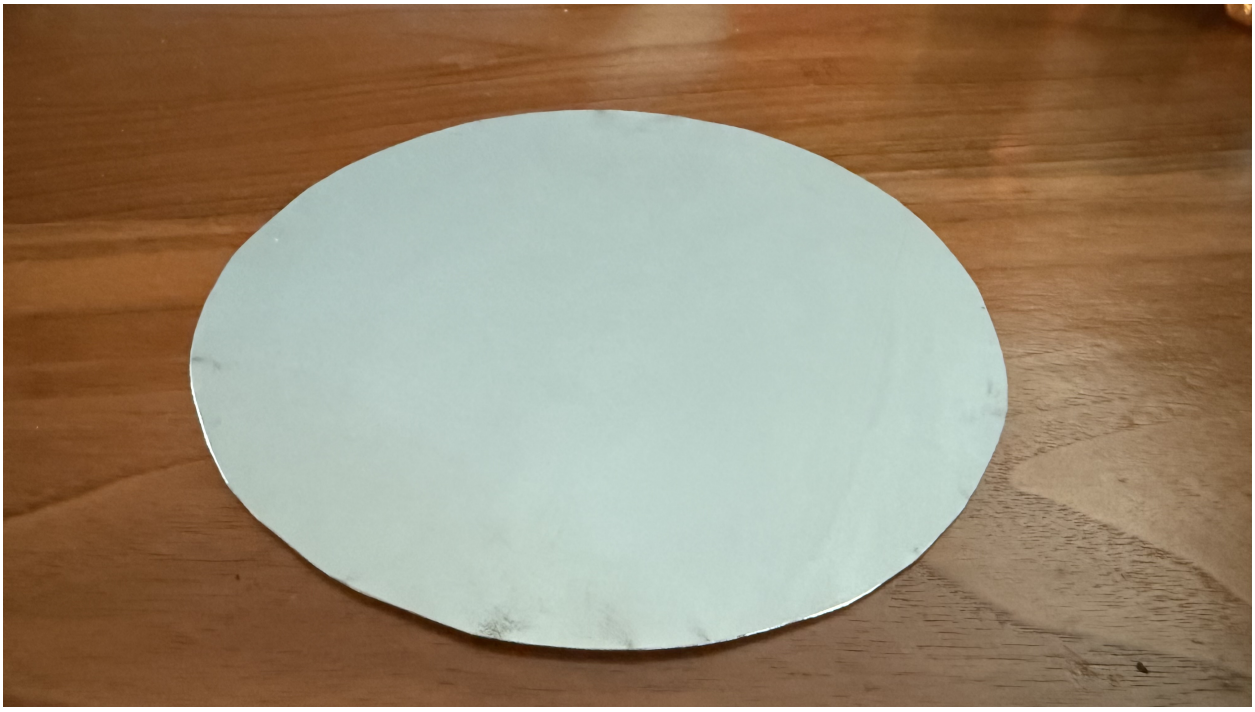

Figure 6.2: Air bearing sheet

Implementing these design considerations, the CAD parts and assembly were fully modeled within SolidWorks. The Figure 6.3 illustrates the base plate assembly with the majority of the components that will be rigidly mounted to the base plate, with a few exceptions for the electrical components and air bearing sheet being included within the model. It primarily features a 3D printed upper base ring designed to house a ball-bearing ring for the inner-gimbal. This base ring is mounted onto the base plate using M4 screws and threaded heat inserts. M4 and M3 threaded inserts are also introduced to the base plate assembly to hold the electrical components: power distribution board, voltage regulator, and PCB. In the center of the base plate, a DS3235 servo is mounted with a bracket that will be used to drive the outer gimbal. A battery compartment is designed to house a 3S Lithium-ion polymer (LIPO) battery with a size of 75 mm x 34 mm x 26.5 mm.

Designing the outer gimbal shown in Figure 6.4, where it will be attached to the servo to the base plate. The outer gimbal sits flush with the ball bearing to provide rigidity and a frictionless rotation. A hole in the center of the gimbal is made for easy when screwing the servo hook down onto the servo. Two arms mounted with M4 screws are used to suspend the inner gimbal where one is modeled to fit with a ball bearing and another used to mount a servo responsible to actuate the inner gimbal.

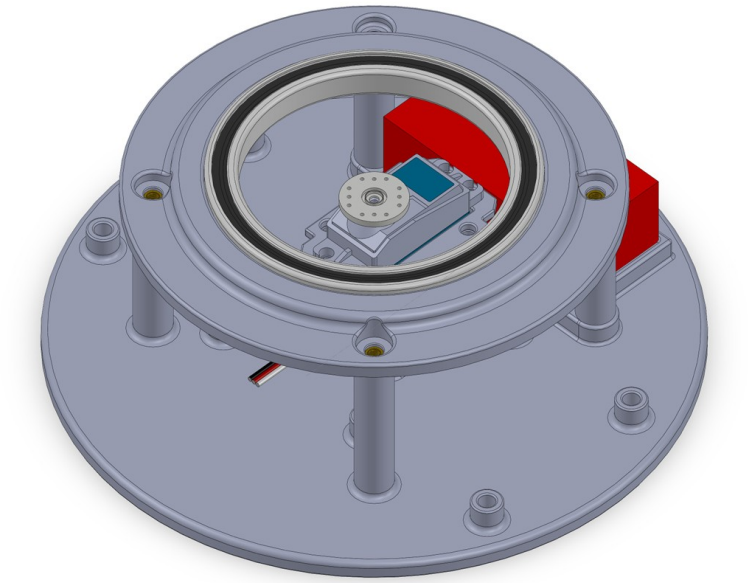

Figure 6.3: Assembly model for baseplate

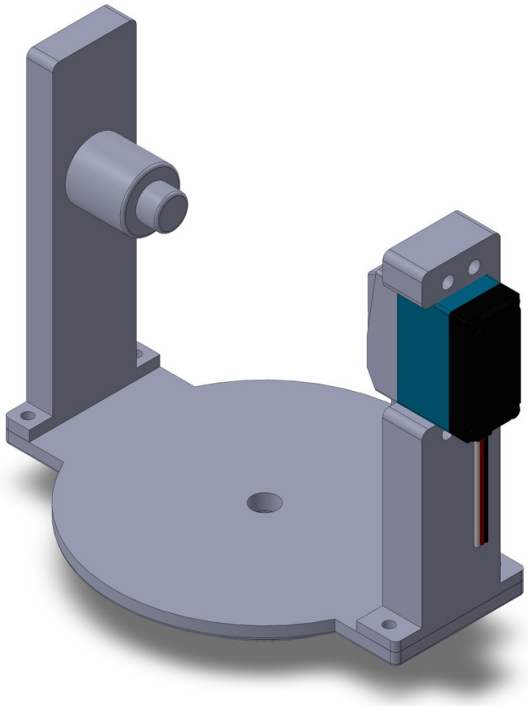

Figure 6.4: Assembly model for outer gimbal

Figure 6.5 illustrates the inner gimbal and two flywheels. As described in the outer gimbal, the inner gimbal consists of an R8-2RS ball bearing and a servo hook to attach to the outer gimbal. There exist 4 M3 hole screws for each BLDC motor to be mounted on. Two slits that extrude-cut through for the wires from the BLDC motors to run through avoid any potential conflict when the DGCMG rotates.

In addition, maximizing the dimensions of the flywheels was set as a design objective to improve the performance of the testbed. Performance is referred to as the system reaching its ability to respond to changes and achieving its desired state. As seen in Chapter 2, the input matrix for the LPV state-space equation for the testbed is proportionally related to the angular momentum built up within the flywheels. In Equation 6.2, enlarging the radius and mass of the flywheel will increase the moment of inertia. Therefore, maximizing the radius and angular velocity of the flywheel would provide the largest angular momentum possible for the testbed, as provided in Equation 6.1.

$$\vec{h} = \vec{\vec{I}}_z \times \vec{\omega} \tag{6.1}$$

$$\vec{\vec{I}}_z = \frac{1}{2} * m_w * \vec{R_w}^2 \tag{6.2}$$

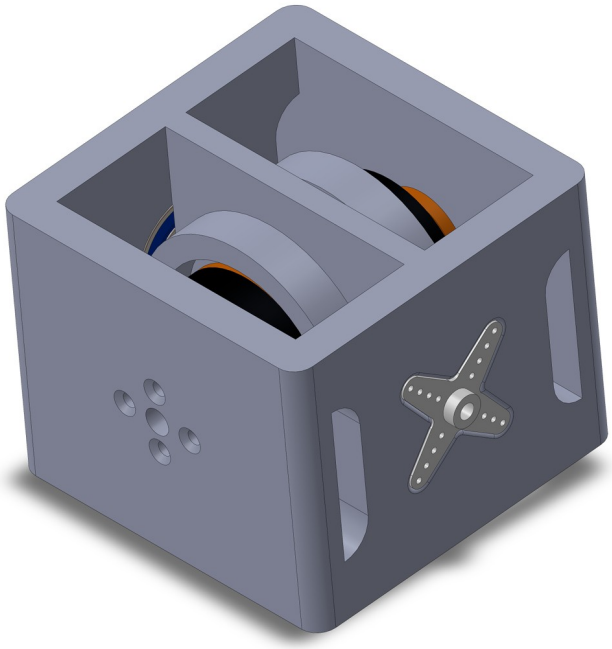

Figure 6.5: Assembly model inner gimbal

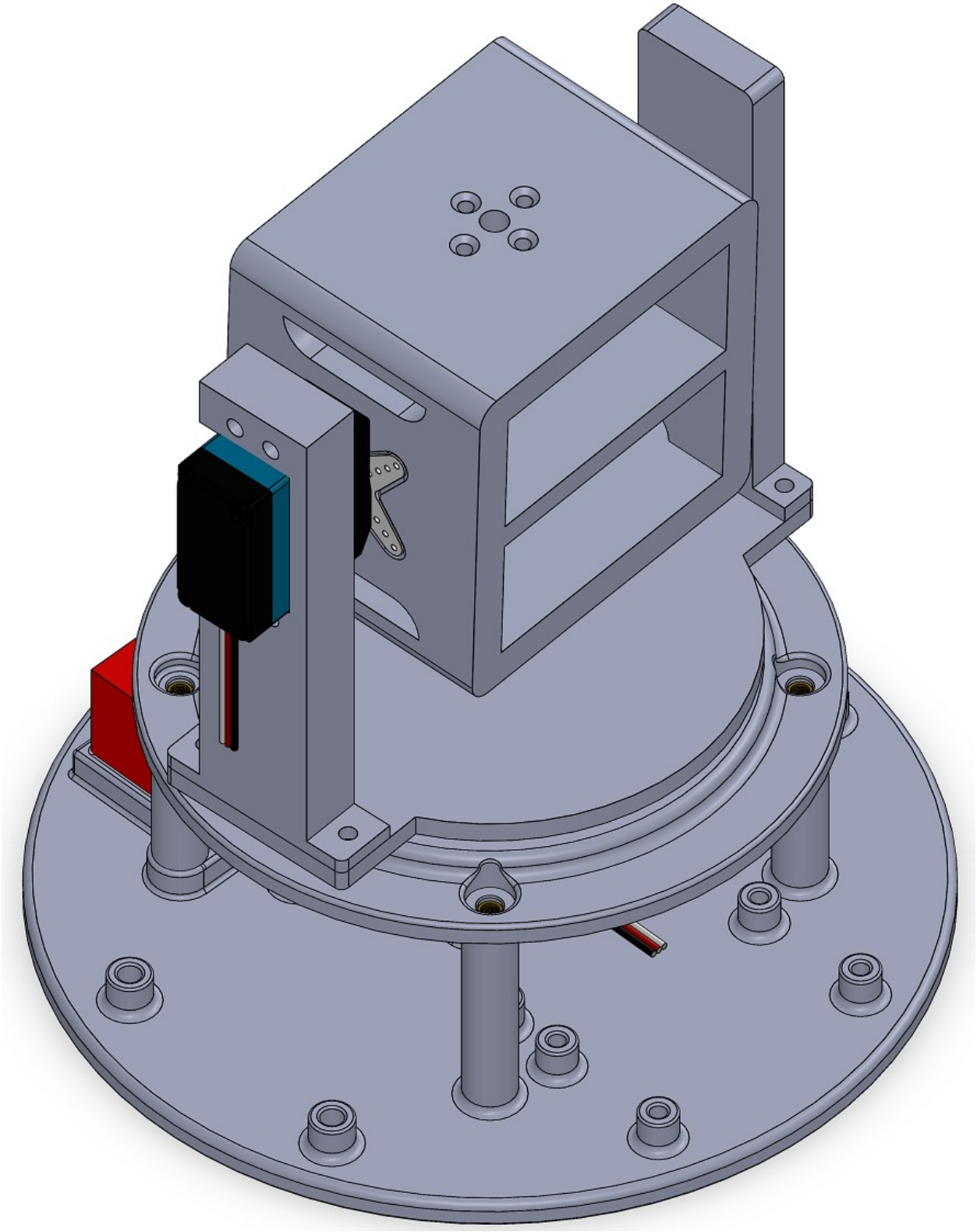

Figure 6.6: Assembly model for testbed

The testbed assembly was manufactured using an additive manufacturing technique known as 3D printing. 6 parts were 3D printed, including: inner gimbal frame, outer gimbal, two outer gimbal sticks, base plate, and servo bracket. Polylactic acid (PLA) was selected as the material to manufacture these parts as it is relatively cost-effective compared to other materials available for 3D printing. Adjustments to the printing temperature were fine-tuned to $195.0°C$ and bed temperature to $50°C$. These temperatures were specifically fine-tuned to reduce possible thermal warping caused by shrinkage from non-uniform heating or cooling. This issue is necessary for troubleshooting the testbed, as the performance of the air bearing can be impaired due to the warping of the base plate. The base plate is vulnerable to warping from having a large flat surface directly on the bed of the printer. Adding an adhesive, such as polyvinyl acetate, to the bed is known to be a common practice in reducing warping and improving adhesion for 3D printing the part. This practice was applied to all 3D printed parts. For the layers of the 3D print, a 0.8mm thickness is used for the top and bottom layer of the print with a 20%. 3D printing settings used for Ultimaker Cura are provided in the Table 6.2.

Table 6.2: Cura settings

Parameter	Value
Layer Height	0.2 mm
Top/Bottom Thickness	0.8 mm
Top Layers	4
Infill Density	20%
Infill Line	Lines
Printing Temperature	$195.0°C$
Build Plate Temperature	$50°C$
Print Speed	89 mm/s
Fan Speed	100%
Build Plate Adhesion Type	Skirt

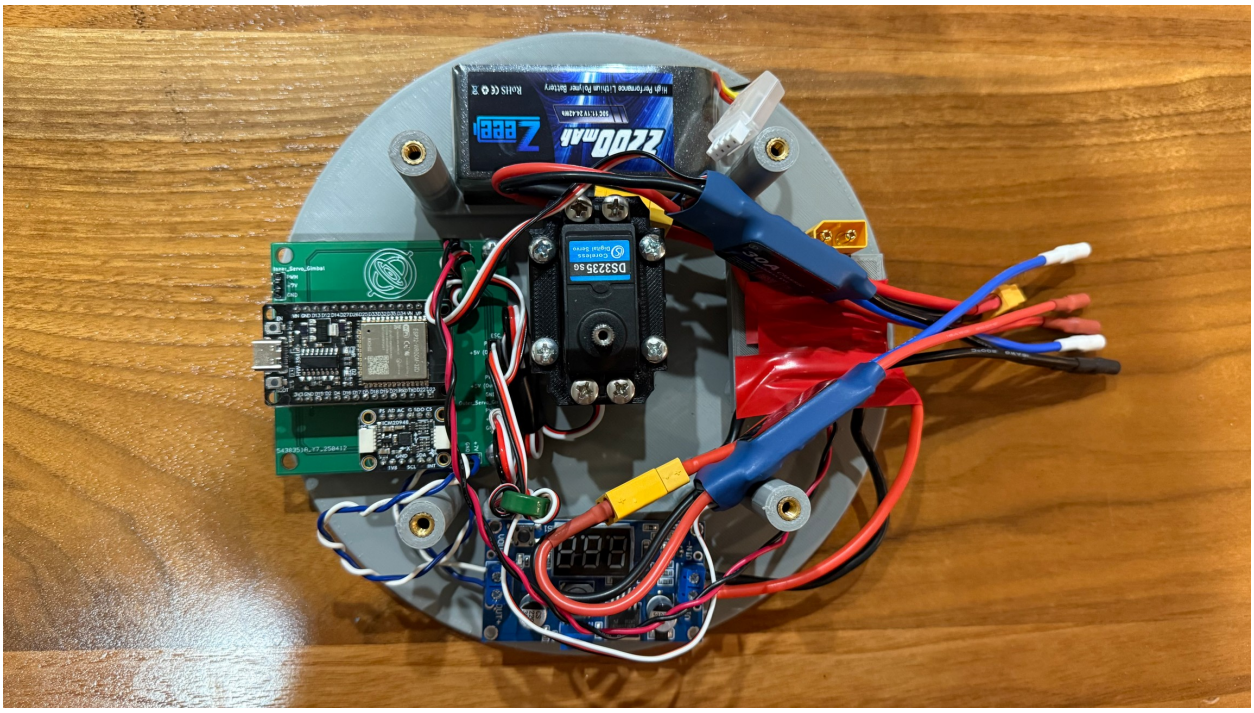

Figure 6.7: Base plate assembly & mounted electronic components

Figure 6.8: Upper base plate assembly

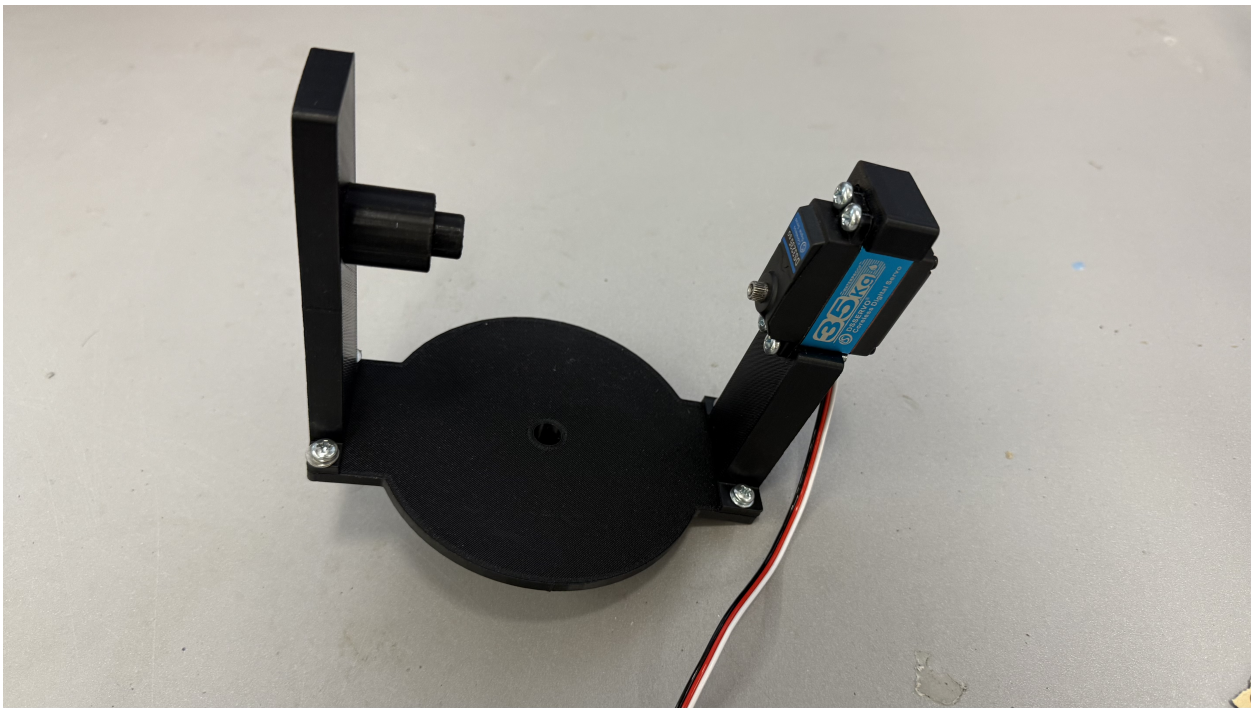

Figure 6.9: Outer gimbal assembly

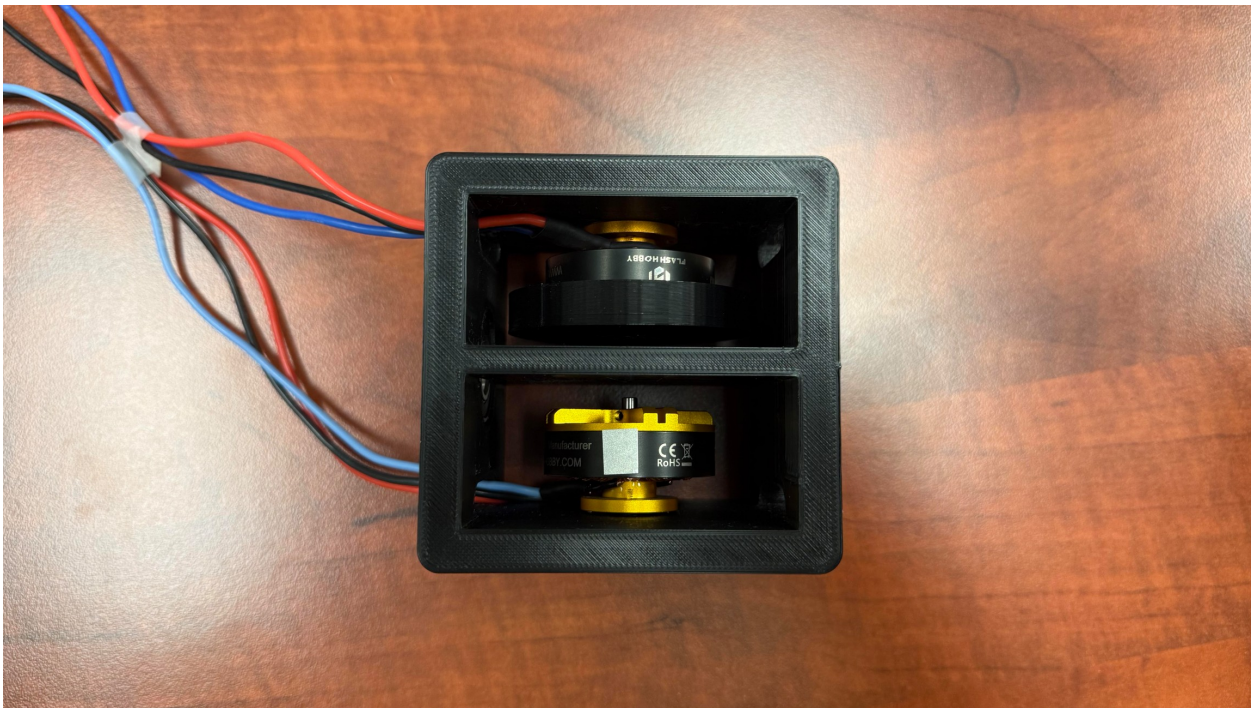

Figure 6.10: Inner gimbal assembly

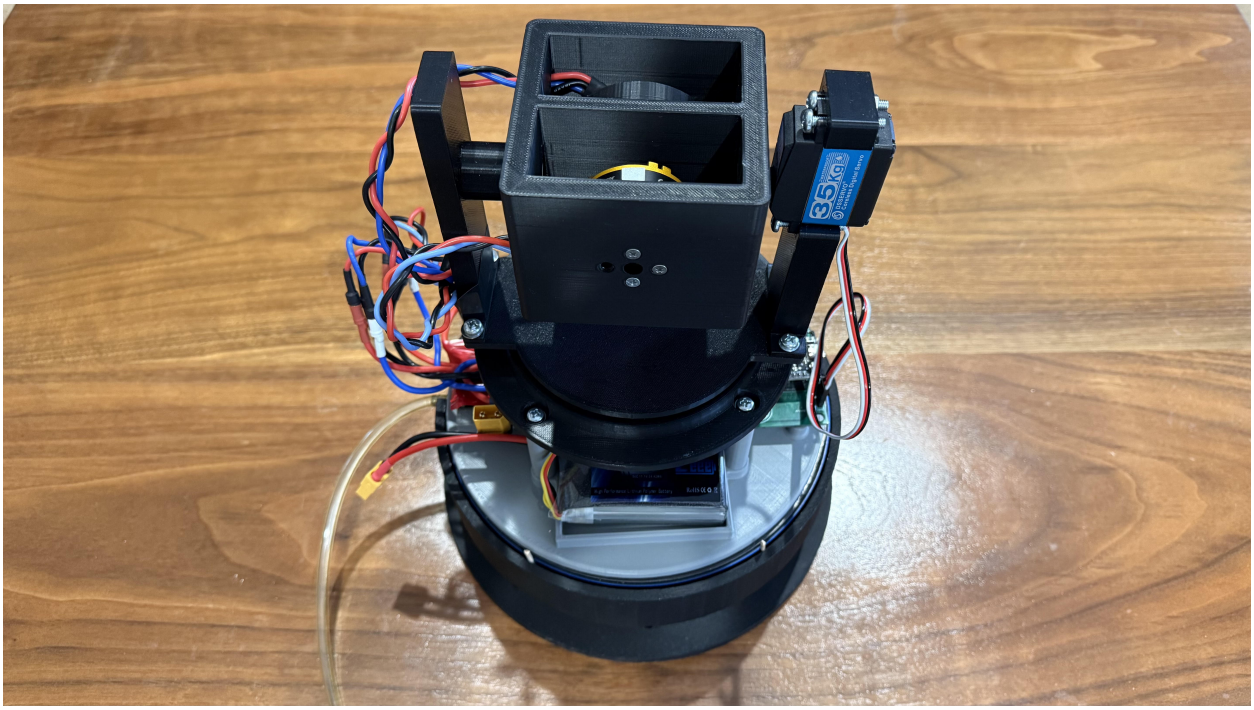

Figure 6.11: Fully testbed assembled

7. Test and Results

7.1 Test Procedures

A documentation for the procedure was created detailing the steps used to run the test. Each step is laid out chronologically, and each step must be followed as listed. These steps are listed in three categories: simulation, pneumatic, and testbed. Each will discuss how to run the simulation, properly set the air bearing, and operate the testbed device. The GitHub repository provides the code required to run the testbed and simulations [45].

Simulation Procedure

1. Using the provided Matlab code in Appendix B paired with Appendix E, input calculations for the angular momentum of the flywheel and moment of inertia of the testbed to obtain the gain matrices.
2. Run Simulink model in Figure 4.5 to simulate the gain-scheduling LQR controller to obtain computational results.

Pneumatic Procedure

3. Check air compressor regulator and ensure that it is set to between 60 to 80 psi for the acceptable range for the New Way air bearing specification.
4. Tightly insert the polyurethane airline into the pneumatic hose coupling of the regulator and onto the air bearing.
5. Turn on the air compressor, then ensure the regulator reaches 60 to 80 psi and air is releasing from the air bearing.

Testbed Procedure

6. Place the testbed above the air bearing and connect the device through USB-C with a computer running the Matlab Simulink model in Figure 5.10.
7. Run Simulink model with the gain matrices in the workspace and enable the device through the initialize button.
8. Input a desired response and let the device perform its maneuver until rest. Ensure safety by avoiding touching the motors during operation.
9. Once completed, turn off the device by re-clicking the initialize button and stopping the MATLAB Simulink model. Export recorded data from the workspace.

Figure 7.1: Test setup

7.2 Results

Calculations for the flywheel angular momentum were performed to obtain the gain variables for the Quasi-LPV LQR model. Exclusion of the flywheel was removed from the BLDC motors. Calculations will only account for the BLDC motors acting as the flywheel for this testbed. Applying Equation 6.1, an approximation for angular momentum can be made, but it is predicted to be lower than the actual value. To obtain the mass properties of the flywheel, SolidWorks is used to help approximate an estimate of the mass and moment of inertia of the rotor. SolidWorks estimates that the mass is 15.73848 g and the moment of inertia is 0.00000285 kg/m^2, knowing that the rotor is constructed using 6082 aluminum bell as material with a known density of 2700 kg/m^3 [15, 48]. A tachometer used to measure the angular velocity of the motor, which topped 755.6 rad/s without load. Plugging these terms into Equation 6.1, the angular momentum of each flywheel is solved to be 0.00215346 $kg \cdot m^2/s$. Thus, the sum of the angular momentum of DGCMG equates approximately to 0.00430692 $kg \cdot m^2/s$. A negative sign was given to indicate that the motor is spinning opposite to the x-axis due to how the motors were aligned and the ESC pins were plugged.

$$\vec{h} = -(0.00000285\ kg/m^2)\ \hat{b_x}\hat{b_x} \cdot (755.6 rad/s)\ \hat{b_x} \tag{7.1}$$

$$\vec{h} = -(0.00215346\ kg \cdot m^2/s)\ \hat{b_x} \tag{7.2}$$

An approximation of the testbed moment of inertia was required before simulations on calculating the gain matrices could be performed. After the 3D prints had been made, each part was weighed to obtain the mass using a scale. Each mass obtained can be used to overwrite the corresponding CAD models in SolidWorks by using the mass properties tab. With this method, SolidWorks can computationally compute an approximation of the moment of inertia for each part and the whole CAD assembly. Because the testbed is only going to rotate about the yaw, a single moment of inertia variable about the Y-axis is required to compute the gain matrices.

Table 7.1: Mass and moment of inertia for testbed component

Parts	Mass [kg]	Moment of Interia (Y-axis) [$kg \cdot m^2$]
Inner Gimbal	.030	0.00275
Outer Gimbal [Base]	.038	0.000054
Outer Gimbal Arm [Servo]	.023	0.000003
Outer Gimbal Arm [Bearing]	.016	0.000002
Bottom Plate	.135	0.000586
Upper Base Ring	.030	0.00011
Servo	0.068	0.000014 & 0.000016
R8-2RS Bearing	0.017	0.000001
6812-2RS Bearing	.155	0.000160
Baseplate Metal Sheet	.121	0.000575
LIPO battery	.030	0.000083
Testbed Assembly	.121	0.00300

Table 7.2: Specified results needed for LPV simulation

Variables	Values
flywheel angular momentum 1 [h_1]	0.00215346 $kg \cdot m^2/s$
flywheel angular momentum 2 [h_2]	0.00215346 $kg \cdot m^2/s$
Testbed Moment of Inertia (Y-axis) [I_y]	0.00300 $kg \cdot m^2$

7.2.1 LPV Gain-Values for Hardware

Similar to in Chapter 3, gain matrices for the LQR must be defined first before any simulations or hardware-in-the-loop can be conducted. The same MATLAB code seen in Appendix B was used to obtain the desired gain matrices required for the gain-scheduling LQR. Changes to the code were made to match the testbed model using the angular momentum of the flywheel and the moment of inertia of the testbed, as seen in Table 7.2. The LQR weight matrices are provided in Table 7.3, which were user-defined. These weight matrices are tuned such that the LQR controller will penalize the yaw position over the yaw rate; thus, the controller will be more responsive to the change of angular velocity. The actuator weight matrix was tuned for the inner servo, as the gyroscopic effects are predominantly more responsive when the servos are at initial conditions.

Table 7.3: Weight matrices for testbed

Gain	Values
State-cost weighted matrix [Q]	$\begin{bmatrix} 50 & 0 \\ 0 & 5 \end{bmatrix}$
Input-cost weighted matrix [R]	$\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$

Computing the code, the gain matrices for the controller have been obtained. Two examples of 4x4 gain matrices are provided, where 7.3 is the gain matrix when the servos are each at -90 degrees. Equation 7.4 starts when the servo positions are approximately near 0 degrees. Equation 7.5 is the gain matrix when the servos reach 90 degrees. Compiling all the LPV LQR gain matrices together, these matrices construct a 10x10 cell array that is used to adjust the gains for different corresponding servo angles.

$$K_{1,1} = \begin{bmatrix} -2.236 & -1.901 \\ 3.122e-16 & 4.653e-17 \end{bmatrix} \tag{7.3}$$

$$K_{5,5} = \begin{bmatrix} 0.388 & -0.330 \\ 2.202 & 1.872 \end{bmatrix} \tag{7.4}$$

$$K_{10,10} = \begin{bmatrix} 2.236 & 1.901 \\ 7.449e-16 & 1.592e-16 \end{bmatrix} \tag{7.5}$$

7.2.2 Cost

Putting the testbed together requires funding, which was granted through the SJSU aerospace engineering department. This funding covered most of the electronic components and parts manufactured. As unveiled in Table 7.4, this table provides all the data on all the necessary components

needed to manufacture the prototype for the testbed, excluding a few unutilized components that became irrelevant or tools used during assembly. For the testbed, the cost for assembling the prototype was a total of $245.83. This cost doesn't account for the cost of specific components provided by SJSU or self-provided. Though the total cost was $245.83, the majority of the cost was saved by having utilized pre-existing components and licensing provided by the department.

Primarily, the device focused on being economically affordable, aimed at academic students or faculty to be capable of manufacturing this device. Achieving a reducing it to a minimal cost was set as the requirement, but having a feasible, functional, and sturdy design requires a trade-off for higher cost in certain aspects: such as the selected motors for their performance, larger bearings to provide rigidness for the outer gimbal, and swapping a MPU6050 sensor for the ICM-20649 for better performance for yaw angle. Because the testbed uses components and licensing from the department, obtaining the components provided by the department or self-provided will significantly add to the cost of the device. Thus, substitution of these components may be considered and is more feasible for larger students.

Table 7.4: Testbed cost

Hardware	Cost	Quantity	Supplier
ESP32	$19.99	1	Amazon
ICM-20649	$14.95	1	Adafruit
Servo	$28.99	2	Amazon
D4215 brushless Motor	$27.99	2	Amazon
ESC 30A	$17.99	2	Amazon
11.1V LIPO Battery	$36.99	1	Amazon
Drone Power Distribution Board XT60	$11.99	1	Amazon
Buck Converter	$12.99	1	Amazon
PCB	$27.89	1	JLCPCB
R8-2RS Bearing	$12.99	1	Amazon
6816-2RS14.09 Bearing	$14.09	1	Amazon
M3 Heat-Set Threaded Inserts	$8.99	1	Amazon
M4 Heat-Set Threaded Inserts	$9.99	1	Amazon
New Way Air bearing	excl.	1	SJSU
Air compressor	excl.	1	Self-provided
Metal sheet	excl.	1	SJSU
3D printer	excl.	1	Self-provided
MATLAB Licensing	excl.	1	SJSU
Total Cost	$245.83		

7.2.3 Testbed Simulation Performance

A simulation was performed on MATLAB Simulink to test the performance and responsiveness of the DGCMG before conducting the test. As seen in Figures 7.2 and 7.3, the simulation focused on testing the trackability with the testbed specifications at 20 degrees. Figure 7.2 displays this feasibility and shows the ability of the system to stabilize the dynamics of the testbed in about 8 seconds. Figure 7.3 shows the input response required for the servos to achieve 20 degrees. Looking at the outer gimbal response, the angle input only needed less than 1 degree. This weak responsiveness is due to the initial conditions and the lack of angular momentum present about the y-axis when the inner servo is at 0 degrees. As the inner gimbal rotates away from its initial position, the gain-scheduling LQR controller is updated to increase the gain responsible for influencing the outer gimbal. The inner gimbals have a larger bias over the outer gimbals at these initial conditions. For the inner servo response, the servo response was observed to act at about 10.5 degrees for the testbed to orient at about 20 degrees. After a short response, the inner servo will return to 0 degrees.

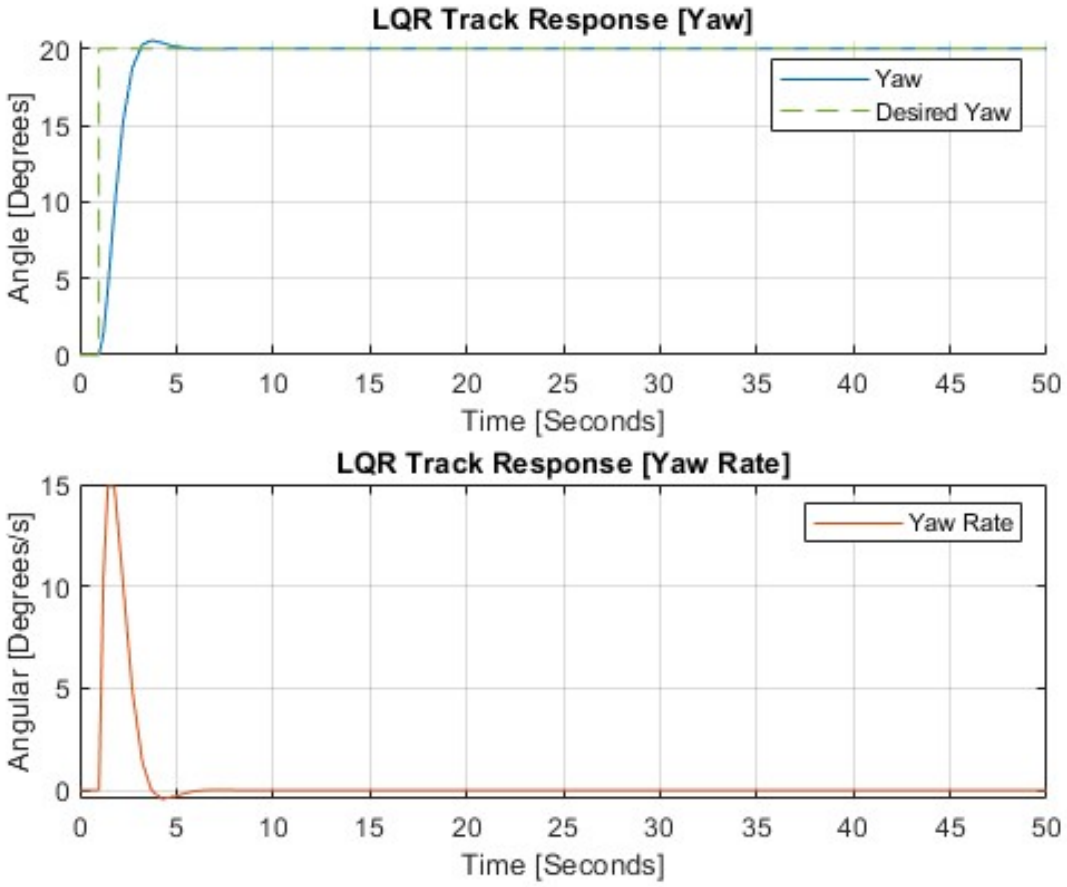

Figure 7.2: LQR-tracking response for simulation using hardware specifications

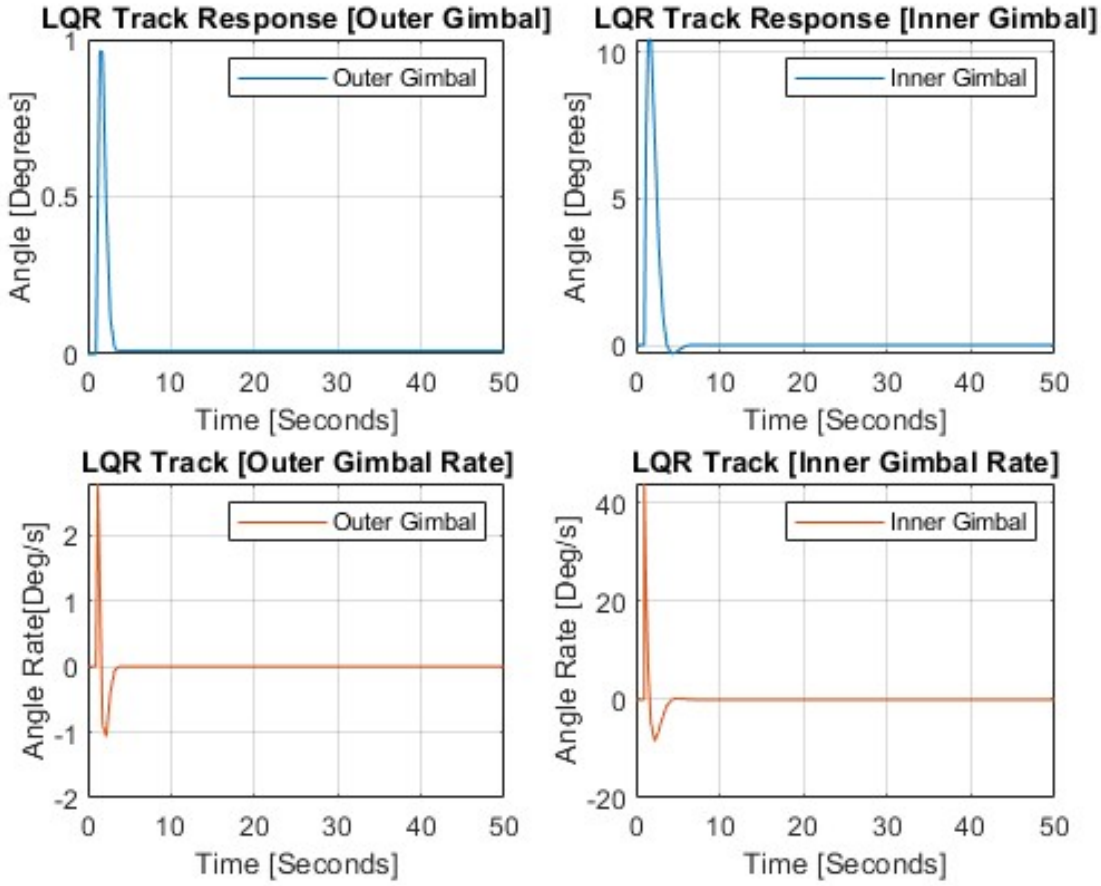

Figure 7.3: LQR-tracking input simulation using hardware specifications

7.2.4 Testbed Performance

Two tests were performed to demonstrate the performance of the hardware and the gain-scheduling LQR controller using hardware-in-the-loop. The first test was conducted to evaluate the performance when the system was perturbed by an external force. With the desired yaw angle set to 0 degrees for initial conditions, a test can be conducted to see if the device can return to its initial angle after an external force perturbed it. Results can be seen in Figure 7.4 and 7.5, which shows a case where the system is perturbed twice at about 18 seconds and 28 seconds into the simulation. These two external forces pushed the testbed in opposite directions to see how the gimbals would have reacted, where the first disturbance to the system was approximately -25.04 degrees and the second pushed 28.7 degrees. The controller responded by actuating the inner gimbal to a 28.4 degree angle, declining steadily back down to a 4.1 degree offset from initial conditions. The second disturbance caused the inner gimbals to actuate down to -32.4 degrees and steadily rise to -12.05 degrees to stabilize the system. Through observations, each disturbance has a cost for the inner servo and will drive it away from initial conditions, leading to saturation over time. The outer gimbal essentially stood at 0 degrees due to the servo angles being near the initial conditions, where the controller prioritizes the inner gimbal as the optimal method in controlling actuation of the system. Through this test, the testbed successfully counteracted the perturbations and stabilized itself

at nearly 0 degrees. Due to having to rely directly on the gyroscope yaw angle, there are periods where the system stabilizes itself back to its initial yaw angle. Still, a noticeable steady offset is present as shown in Figure 7.4. This issue is likely tied back to assembly when the magnetometers lose calibration during the soldering process with the PCB board.

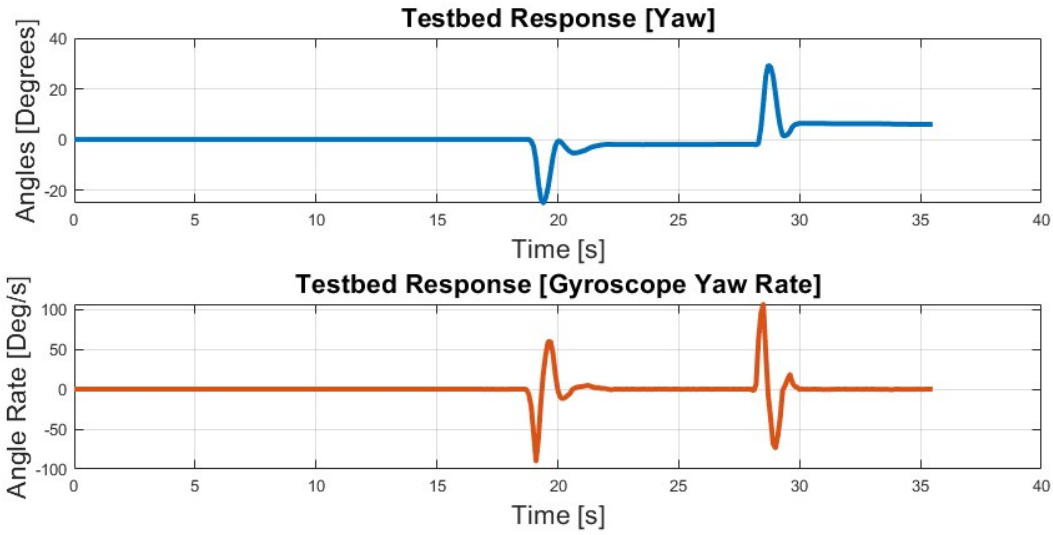

Figure 7.4: Testbed yaw response

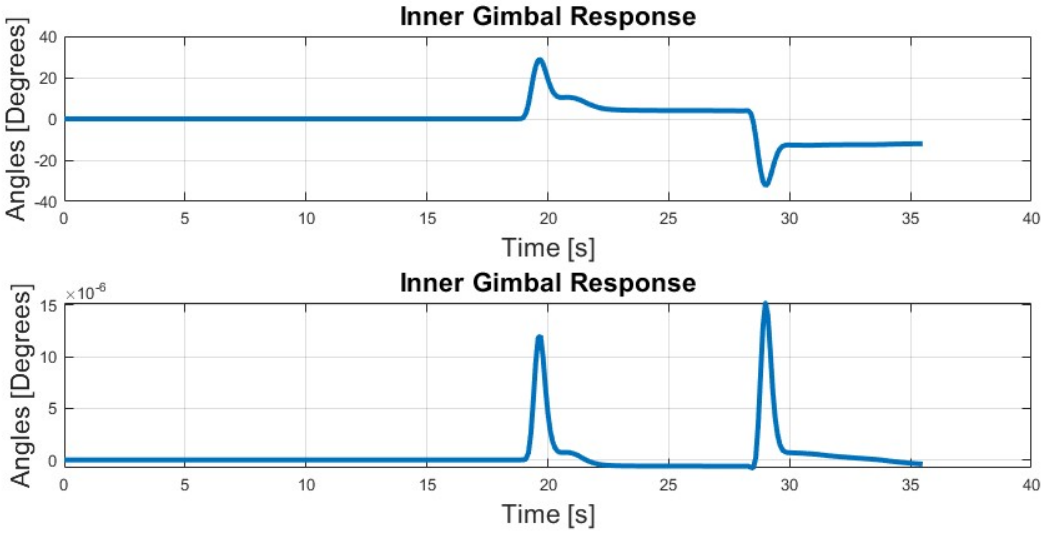

Figure 7.5: Testbed servo input response

The second test focused on the trackability of the system. A similar test to the simulation was given that tested the performance of the system to track 20 degrees after boot-up. With this test, the testbed was evaluated for its performance using the improved trackability method on the controller. Testbed results can be seen in Figure 7.7 and 7.6 that test a single desired state for 20 degrees at 14 seconds. In this case, the bias of the inner and outer gimbals shares behaviors

similar to those of the perturbation test. A 16.88 degree change was made on the inner servo and was driven down to a steady 5.58 degrees. The outer servo stayed at 0 degrees, with negligible response. The effect that this maneuver had on the testbed is seen in Figure 7.6, where the testbed quickly rotated itself to about 18 degrees and reached steady state approximately 3 seconds after the maneuver was performed. Interestingly, the testbed seemed to be 2 degrees under the desired angle of 20 degrees. Though not perfectly, the controller was proven capable of tracking a desired state to a certain extent.

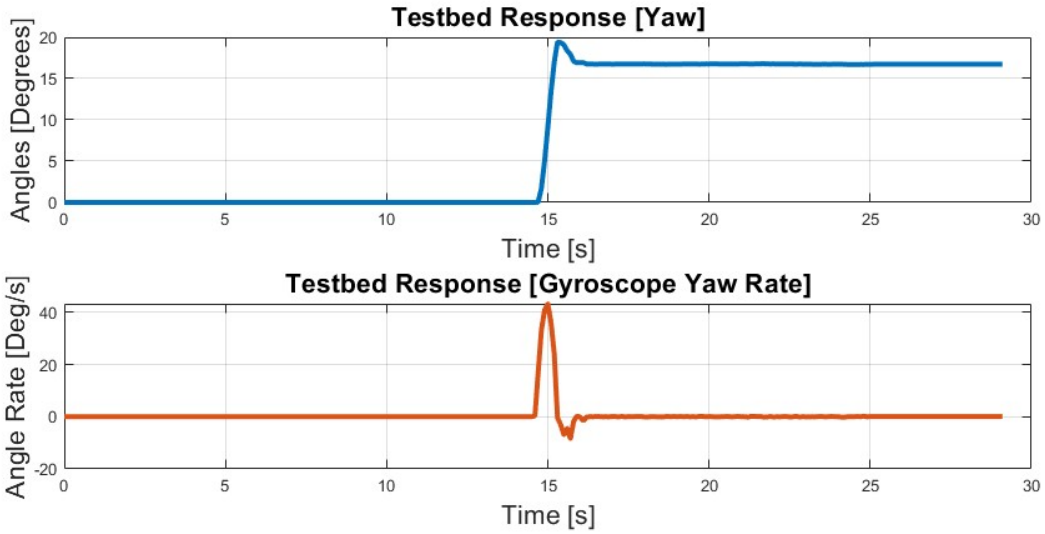

Figure 7.6: Testbed response for tracking

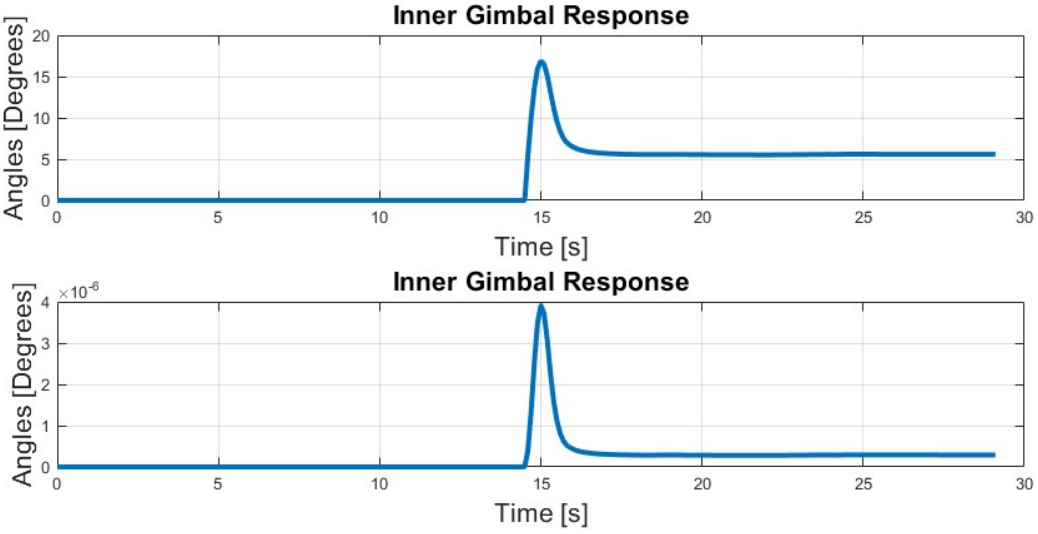

Figure 7.7: Testbed servo input response for tracking

7.3 Discussion

7.3.1 Analysis on Testbed Performance

The purpose of having the simulation was to initially obtain the gain matrices to employ the gain-scheduling techniques in the DGCMG. Running a simulation also provides data that would help predict the response the testbed is expected to exhibit, as shown in Figures 7.2 and 7.3. In this result, the SIMO system and the gain-scheduling LQR controller were demonstrated to be feasible, and the performance of the testbed was evaluated in simulation. As a callback to the results of the simulation, the actuation of the inner servo moves for 0 to 10 degrees at the rate of 40 degrees/s in under 12 seconds. The response it had on the system was capable of tracking it towards the desired reference point of 20 degrees. Thus, the gyroscopic torque exhibits a fast responsiveness behavior when rotating about the yaw angle.

Evaluating the simulation and testbed results together, a comparison can be made to examine how these two models perform under the same conditions. Looking at the two results, the gyroscopic effects of the simulation matched the exact behavior of the testbed during its test. In each of the responses were able to track the desired angle closely. Both were shown to resolve the maneuvers within 5 seconds around similar amounts of time. A difference that was made was the actuation cost of the inner gimbal, where the simulation gimbal cost 10.5 degrees while the experiment required around 17 degrees. With a 6.5 degree difference between actuation costs, it was anticipated that the model may vary in response compared to the actual physical model. To improve the accuracy of the physical model, adjusting the calculations of the flywheel moment of inertia and the moment of inertia could help improve the performance of the system. Comparing these cases, validations of the testbed were made as the physical behaviors aligned with the simulation data. Both respond similarly when a maneuver is executed and support the predicted dynamic of the system. In addition, both show the same bias for the inner and outer gimbal. Verifications of the performance of the two models could be considered to closely match, as results still match in terms of yaw response, but the actuation cost of the inner gimbal is higher in the experimental test.

7.3.2 Analysis on Hardware

During the tests, hardware behavior was monitored to ensure that the setup of the system satisfies the functionality and requirements for the device. For the pneumatic system, the basic setup for the air bearing performance exceeded initial expectations. Though the air bearing specifications for the required PSI were 60 to 80 psi, it was operational well below 60 psi with little performance loss. This allows for longer tests without the frequent need to activate the compressor. To constrain the air bearing to only having rotational motion and removal of linear motion, thin pegs were installed between the air bearing 3D printed shell and the air bearing housing. This operated well, though it minimally impacted the air bearing performance by adding friction caused by the contact with the base plate and pegs.

For the electronics, each operated as designed. All electronics were powered, and the LIPO battery was sufficient to support the system for a long operation time. The BLDC motors meet the requirements for angular velocity, spinning around 755.6 rad/s. The position servos also satisfy the requirements and sustain the requirements for gimbal rate when operating at +7V. One caveat is that the IMU lost the calibration for the magnetometer after the PCB was soldered onto the PCB. IMU can be recalibrated following the steps in Chapter 5.

Given that the system performed directly with MATLAB Simulink, much of the performance and limitations on what the testbed can do lie directly in how the hardware and software interact with each other. The built-in functions, such as lqr() or the lookup table in Simulink, made the development of the control algorithm to enhance the manageability of troubleshooting issues. Simulink did have some challenges that involved developing the PWM signals for the ESC, calibrating IMU data, preventing exceeding missing ticks, and timing between the ESP32 and Simulink interface. Once those challenges were settled, the Simulink model could be easily altered and fine-tuned.

7.3.3 Potential Performance Impacts from Testbed Artifacts

Observing the testbed, noticeable unexpected phenomena have been observed that affected the testbed performance. One such example was spotted as servo jitter, which would cause the servos to essentially shake when executing their maneuvers. Noticeable jitters are induced by the controller itself, possibly caused by the direct commands to control the position of the servos through integration methods using the gimbal rates. A solution was to properly fine-tune the state weight matrix, adjusting the diagonal accordingly. This has been done through small increments through trial and error. Negligently over-tuning the state weight matrix can further increase the jitter effect. An additional method to help reduce the jitter further is to decrease the angular momentum of the flywheel when solving for the gain matrices. Doing so can increase the input response of the actuator to be more reactive while keeping the sensitivity minimal when integrating angular position.

Additionally, having the integration method on the gimbal rates causes an integration that may exceed the gimbal angle limit, thus causing potential issues, particularly to the gain-scheduling controller, as the search-up tables aren't designed to cover those bounds of ± 90 degrees. A possible solution to this can be to develop a MATLAB function block within Simulink to saturate the limit of the gimbal position during integration. The function will have to handle integration using the post-saturated position. Furthermore, an additional effect happens when the inner gimbal becomes saturated near ± 90 degrees. When the gimbal becomes saturated, the controller computes that the only actuation available is the outer gimbal; thus, the inner gimbal potentially becomes gimbal-locked. There are occasional instances where the inner gimbal is displaced from the immobilized state, but generally only for a short time. This phenomenon is not directly a flaw, as the controller and dynamics of the system were expected. To reduce this effect from happening, the inner gimbal could be limited to angles where both inner and outer gimbals can still have effects.

8. Conclusion & Future Works

8.1 Conclusion

To conclude this study, the project developed a testbed for a double-gimbal control moment gyroscope that aims to simulate the key aspects observed in spacecraft dynamics. The development of this device draws from various aspects of mechatronics, making it a multidisciplinary effort involving fields of control systems, software, and electrical engineering. All together, this device demonstrated performance in tracking and stabilization through the utilization of the gain-scheduling LQR. Validations and verification of the physical device were confirmed by comparing its behavior with the equations of motion and simulating the nonlinear dynamics using quasi-LPV models. Both exhibited similar dynamical behaviors and controller responses. Though the accuracy of the model was slightly off, it was anticipated that assumptions and uncertainties were expected to deviate from the physical system.

In hopes, this project is meant to bring inspiration to future STEM students at San Jose State University. Essentially allowing students to conjure their creations into reality with the academic knowledge they inhered. It is planned to bring insight for them by providing documentation on the development of the project, while providing a device that will allow for accessibility for future tests and improvements. The testbed incorporates MATLAB for controls, enabling engineering students to readily develop their own controllers and integrate it with the testbed; particularly beneficial for aerospace students who are familiar with MATLAB. Ultimately, this project serves as the paved foundation for those who push forward with this project and as a catalyst for those who spiral into a project of their own path.

8.2 Future Work

To improve the design of the project, future implementations are provided to further enhance the DGCMG and remove its flaws. One significant change that can benefit the design is by changing the position servos to a continuous servo with a readable potentiometer integrated within it. As the continuous servos control the speed, this alteration will remove the need to integrate the gimbal rate for position on the current design. Doing so should remove the jitter and improve the smoothness of the gimbals. The potentiometer will provide direct feedback on the position of the servo for the look-up tables and LPV model for the controller.

Furthermore, other mentions on the physical design can be improved on. Such as substituting for an air bearing designed specifically for rotation to reduce friction, shifting the outer gimbal to lie on the y-axis to possibly test further configurations, and implementing a slip-ring to reduce the cables on the BLDC motors.

Lastly, potential improvements to the software would enhance the functionality and performance of the system. Having a function with Bluetooth would essentially remove the need for the USB-C cable and thus remove the possible tension induced. Additionally, improvements on the current design of the control can always be made either through testing the simulation with noise, fine-tuning the system, or improving methods for obtaining system specifications to help improve

the accuracy of the models. The model of the LPV model can be improved on as well to account for the moment of inertia of the gimbals as it rotates. Accounting for possible frictional forces or air resistance on the model will add the damping present on the physical testbed.

References

[1] Pereira, H., Loureno, P., and Batista, P., "From singularity analysis to singularity avoidance: novel metric and convex allocation for spacecraft attitude control with control moment gyros," *Acta Astronautica*, Vol. 225, 2024, pp. 41–54.
https://doi.org/10.1016/j.actaastro.2024.08.047.

[2] Gang, L., Dengyun, W., Ming, L., Lin, L., Xing, T., and Jiyang, Z., "Research on high accuracy, long life, and high-reliability technique of control moment gyroscope," *IEEE International Conference on Mechatronics and Automation*, 2016, pp. 1536–1540.
https://doi.org/10.1109/ICMA.2016.7558792.

[3] Prabhakaran, V. S., and Sanyal, A., "Adaptive Singularity-Free Control Moment Gyroscopes," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 11, 2018, pp. 2416–2424.
https://doi.org/10.2514/1.G003545.

[4] Papakonstantinou, C., Lappas, V., and Kostopoulos, V., "A Gimballed Control Moment Gyroscope Cluster Design for Spacecraft Attitude Control," *Aerospace*, Vol. 8, No. 9, 2021.
https://doi.org/10.3390/aerospace8090273.

[5] Sheerin, T., "Design and utility assessment of attitude control systems for EVA task performance," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, January 2015.

[6] Siahpush, A., and Sexton, A., "A study for semi-passive gravity gradient stabilization of small satellites," *Globesat, Inc.*, 1987.

[7] Berner, R., "Control moment gyro actuator for small satellite applications," Ph.D. thesis, Stellenbosch: University of Stellenbosch, Stellenbosch, South Africa, 2005.

[8] Zorita, J., *Dynamics of small satellites with gravity gradient attitude control*, KTH Electrical Engineering, Stockholm, Sweden, 2011.

[9] Zhou, J., *Attitude Determination and Control of the CubeSat MIST*, KTH Vetenskap OCH Konst, Stockholm, Sweden, 2016.
`https://api.semanticscholar.org/CorpusID:114329773`.

[10] N.A, "DS Servo HV Coreless Digital," , n.d..
`https://hajim.rochester.edu/me/sites/kelley/me240/DS3235-270_datasheet.pdf`.

[11] Siepert, B., "Adafruit ICM20649 Wide-Range 6-DoF IMU Accelerometer and Gyro," *Adafruit Industries*, 2024.
`https://cdn-learn.adafruit.com/downloads/pdf/adafruit-icm20649-wide-range-6-dof-imu-accelerometer-and-gyro.pdf`.

[12] N.A, "200mm Flat Round Air Bearing," *New Way Air Bearing*, n.d..
https://www.newwayairbearings.com/catalog/product/200mm-flat-round-air-bearings/.

[13] N.A, "ESP32 Pinout Reference," *Last Minute Engineers*, n.d.
https://lastminuteengineers.com/esp32-pinout-reference/.

[14] N.A, "30A RC Brushless Motor Electric Speed Controller ESC 3A UBEC with XT60 3.5mm bullet plugs," *RC Electical Parts*, n.d.
https://www.rcelectricparts.com/30a-esc---classic-series.html.

[15] N.A, "D4215 Brushless DC Motor," *Flash Hobby*, n.d.
https://www.flashhobby.com/d4215-brushless-dc-motor.html.

[16] Chubb, W. B., Kennel, H. F., Rupp, C. C., and Seltzer, S. M., "Flight Performance of Skylab Attitude and Pointing Control System," *Journal of Spacecraft and Rockets*, Vol. 12, No. 4, 1975, pp. 220–227.
https://doi.org/10.2514/3.56967.

[17] Camps, A., "Nanosatellites and Applications to Commercial and Scientific Missions," *Satellites Missions and Technologies for Geosciences*, edited by V. Demyanov and J. Becedas, IntechOpen, Rijeka, 2019, Chap. 9.
https://doi.org/10.5772/intechopen.90039.

[18] Gaude, A., and Lappas, V., "Design and Structural Analysis of a Control Moment Gyroscope (CMG) Actuator for CubeSats," *Aerospace*, Vol. 7, No. 5, 2020.
https://doi.org/10.3390/aerospace7050055.

[19] Votel, R., and Sinclair, D., "Comparison of Control Moment Gyros and Reaction Wheels for Small Earth-Observing Satellites," *26th AIAA/USU Conference on Small Satellites*, , No. 1316, August 2012, p. 17.
https://doi.org/10.2514/1.G003545.

[20] Leve, F., Hamilton, B., and Peck, M., *Spacecraft Momentum Control Systems*, Space Technology Library, Springer International Publishing, Gewerbestrasse 11, Switzerland, 2015.
https://doi.org/10.1007/978-3-319-22563-0_2.

[21] Roser, X., and Sghedoni, M., "Control Moment Gyroscopes (CMG's) and their Application in Future Scientific Missions," *Spacecraft Guidance, Navigation and Control Systems, Proceedings of the 3rd ESA International Conference*, 1997, p. 523.

[22] Dearing, T. L., Hauser, J., Petersen, C., Nicotra, M. M., and Chen, X., "Attitude Trajectory Optimization and Momentum Conservation with Control Moment Gyroscopes," *IFAC-PapersOnLine*, Vol. 56, No. 2, 2023, pp. 1937–1943.
https://doi.org/10.1016/j.ifacol.2023.10.1085.

[23] Lungu, M., "Control of double gimbal control moment gyro systems using the backstepping control method and a nonlinear disturbance observer," *Acta Astronautica*, Vol. 180, 2021, pp.

639–649.
https://doi.org/10.1016/j.actaastro.2020.10.040.

[24] Li, H., Zheng, S., and Ning, X., "Precise Control for Gimbal System of Double Gimbal Control Moment Gyro Based on Cascade Extended State Observer," *IEEE Transactions on Industrial Electronics*, Vol. 64, No. 6, 2017, pp. 4653–4661.
10.1109/TIE.2017.2674585.

[25] Stevenson, D., and Schaub, H., "Nonlinear Control Analysis of a Double-Gimbal Variable-Speed Control Moment Gyroscope," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 3, 2012, pp. 787–793.
https://doi.org/10.2514/1.56104.

[26] Nagabhushan, V., Fitz-Coy, N. G., and Leve, F. A., "Attitude Control System For Small Satellites," *University of Florida Research Foundation, Inc.*, 2015.

[27] Wie, B., *Space Vehicle Dynamics and Control*, AIAA education series, American Institute of Aeronautics and Astronautics (AIAA), Washington, DC, 1998.
`https://books.google.com/books?id=n97tEQvNyVgC`.

[28] Creaser, C., and Bauer, R., "Logic-based gimbal compensator for the double-gimbal scissored-pair control moment gyroscope using magnetic torquers," *Advances in Space Research*, Vol. 73, No. 1, 2024, pp. 254–270.
https://doi.org/10.1016/j.asr.2023.10.023.

[29] Yoon, H., and Tsiotras, P., "Singularity Analysis of Variable Speed Control Moment Gyros," *Journal of Guidance Control and Dynamics*, Vol. 27, 2004, pp. 374–386.
`https://api.semanticscholar.org/CorpusID:120396652`.

[30] WU, Y., HAN, F., HUA, B., CHEN, Z., and YU, F., "Singularity analysis for single gimbal control moment gyroscope system using space expansion method," *Chinese Journal of Aeronautics*, Vol. 31, No. 4, 2018, pp. 782–794.
https://doi.org/10.1016/j.cja.2018.01.022.

[31] Jung, D., and Tsiotras, P., *An Experimental Comparison of CMG Steering Control Laws*, AIAA/AAS Astrodynamics Specialist Conference and Exhibit, 2004, p. 5294.
https://doi.org/10.2514/6.2004-5294.

[32] Paradiso, J. A., "Global steering of single gimballed control moment gyroscopes using a directed search," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 5, 1992, pp. 1236–1244.
https://doi.org/10.2514/3.20974.

[33] Kojima, H., Nakamura, R., and Keshtkar, S., "Steering control law for double-gimbal scissored-pair CMG," *Advances in Space Research*, Vol. 66, No. 4, 2020, pp. 771–784.
https://doi.org/10.1016/j.asr.2020.05.007.

[34] Seo, H.-H., Bang, H., and Cheon, Y.-J., "Steering law of control moment gyros using artificial potential function approach," *Acta Astronautica*, Vol. 157, 2019, pp. 374–389. https://doi.org/10.1016/j.actaastro.2018.05.054.

[35] Bloemers, T., and Toth, R., *Equations of motion of a control moment gyroscope*, Tu/e University of Technology, 2019.

[36] Xu, Z., Pan, S., Chen, L., Lu, M., Liang, Z., and Qin, Y., "Dynamic modeling and open-loop analysis of a control moment gyroscope considering the influence of a flexible vibration isolator," *Mechanical Systems and Signal Processing*, Vol. 169, 2022, p. 108611. https://doi.org/10.1016/j.ymssp.2021.108611.

[37] Toriumi, F., and Anglico, B., "Lyapunov-Based Nonlinear Control Applied to a Control Moment Gyroscope," *Journal of Control, Automation and Electrical Systems*, Vol. 32, 2021. https://doi.org/10.1007/s40313-020-00681-x.

[38] White, A. P., *Linear parameter-varying control for engineering applications*, SpringerBriefs in electrical and computer engineering, Control, automation and robotics, Springer, London, 2013.

[39] Mohammadpour, J., and Scherer, C. W., *Control of Linear Parameter Varying Systems with Applications*, Springer, Boston, MA, 2012. https://doi.org/10.1007/978-1-4614-1833-7.

[40] Bett, C. J., "9 - Gain-Scheduled Controllers," *The Electrical Engineering Handbook*, edited by W.-K. CHEN, Academic Press, Burlington, 2005, pp. 1107–1114. https://doi.org/10.1016/B978-012170960-0/50086-4.

[41] Neves, G. P., Anglico, B. A., and Oliveira, R. C., "H2 gain-scheduled state-feedback design with experimental validation in a control moment gyroscope represented as a polynomial LPV model," *Mechatronics*, Vol. 89, 2023, p. 102917. https://doi.org/10.1016/j.mechatronics.2022.102917.

[42] Pinheiro, E. R., and de Souza, L. C. G., "Design of the Microsatellite Attitude Control System Using the Mixed H2/H Method via LMI Optimization," *Mathematical Problems in Engineering*, Vol. 2013, No. 1, 2013, p. 257193. https://doi.org/10.1155/2013/257193.

[43] Kanata, S., Hayashida, F., and Shimomura, T., "Experimental Verification for Attitude Control of Spacecraft Utilizing Double-Gimbal Variable-Speed Control Moment Gyroscopes," *Transactions of the Japan Society for Aeronautical and Space Sciences*, Vol. 66, No. 6, 2023, pp. 209–216. https://doi.org/10.2322/tjsass.66.209.

[44] Mashtakov, Y., Tkachev, S., and Ovchinnikov, M., "Use of External Torques for Desaturation of Reaction Wheels," *Journal of Guidance, Control, and Dynamics*, Vol. 41, 2018. 10.2514/1.G003328.

[45] Nguyen, J., "Control Moment Gyroscope SJSU," , May 2025. 10.5281/zenodo.15377136, https://github.com/TeslaHead/Control-Moment-Gyroscope-SJSU.

[46] Tedaldi, D., Pretto, A., and Menegatti, E., "A robust and easy to implement method for IMU calibration without external equipments," 2013. 10.1109/ICRA.2014.6907297.

[47] Jayakody, D., and Sucharitharathna, K., "Control Unit for a Two-Wheel Self-Balancing Robot," *Global Journal of Researches in Engineering*, 2019, pp. 7–12. 10.34257/GJREJVOL19IS1PG7.

[48] N.A, "Aluminum 6082-T6," *MatWeb*, n.d. https://www.matweb.com/search/datasheet_print.aspx?matguid=fad29be6e64d4e95a241690f1f6e1eb7.

Appendix A — Deriving Equations of Motion for Spacecraft with Double Control Moment Gyroscope

Starting with Newton's second law for rotational bodies

$$\frac{d}{dt}\vec{H}_o = \vec{M}_o \tag{A.1}$$

Expanding LHS to express the total angular momentum of the satellite

$$\dot{\vec{H}}_s + \vec{\omega}^{\times} \times \vec{H}_s = \vec{T}_{ext} \tag{A.2}$$

Where $\omega^{\times}$ is the skew-matrix

$$\vec{\omega}^{\times} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{A.3}$$

Spacecraft momentum is expressed as

$$\vec{H}_s = \vec{\vec{I}} \times \vec{\omega} + \vec{h} \tag{A.4}$$

Substituting A.4 into A.2 obtains A.5

$$(\vec{\vec{I}} * \dot{\vec{\omega}} + \dot{\vec{h}}) + \vec{\omega} \times (\vec{\vec{I}} * \vec{\omega} + \vec{h}) = \vec{T}_{ext} \tag{A.5}$$

Leaving the 1st derivative term of the body angular velocity

$$(\vec{\vec{I}} * \dot{\vec{\omega}}) = -\vec{\omega}^{\times} \times (\vec{\vec{I}} * \vec{\omega}) - \dot{\vec{h}} + \vec{T}_{ext} \tag{A.6}$$

Introducing the cosine matrix tables responsible for representing the rotation of the flywheel to the spacecraft body frame. R_{γ_1} is used to rotate about the y-axis of ny. R_{γ_2} is used to rotate about the z-axis of ay.

$$R_y(\gamma_1) = \begin{bmatrix} \cos\gamma_1 & 0 & -\sin\gamma_1 \\ 0 & 1 & 0 \\ \sin\gamma_1 & 0 & \cos\gamma_1 \end{bmatrix} \tag{A.7}$$

$$R_z(\gamma_2) = \begin{bmatrix} \cos\gamma_2 & -\sin\gamma_2 & 0 \\ -\sin\gamma_2 & \cos\gamma_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.8}$$

Cross multiplication can be applied to obtain a cosine matrix to get from b-frame to n-frame.

$$R(\gamma_1,\gamma_2) = R_z(\gamma_2)R_y(\gamma_1) = \begin{bmatrix} \cos\gamma_2\cos\gamma_1 & \sin\gamma_2 & -\cos\gamma_2\sin\gamma_1 \\ -\sin\gamma_2\cos\gamma_1 & \cos\gamma_2 & \sin\gamma_2\sin\gamma_1 \\ \sin\gamma_1 & 0 & \cos\gamma_1 \end{bmatrix} \tag{A.9}$$

The moment of inertia about the principal axis of the spacecraft shown as

$$\vec{\vec{I}} = \begin{bmatrix} Ix & 0 & 0 \\ 0 & Iy & 0 \\ 0 & 0 & Iz \end{bmatrix} \tag{A.10}$$

The total angular momentum of two flywheels can be expressed in A.11 where its expressed in $\hat{b_x}$ frame.

$$\vec{h} = h_1 + h_2 \tag{A.11}$$

To take the derivative of angular momentum, the Jacobian matrix multiplied by the gimbal rate is obtained.

$$\dot{h}(\gamma) = \frac{dh(\gamma_{1,2})}{dt} = \frac{dh(\gamma_{1,2})}{d\gamma} * \frac{d\gamma}{dt} = G(\gamma_{1,2}) * \dot{\gamma} \tag{A.12}$$

$$G(\gamma) = \begin{bmatrix} \frac{\partial f(\gamma)}{\partial \gamma_1} & \frac{f(\gamma)}{\partial \gamma_2} \\ \vdots & \vdots \end{bmatrix} \tag{A.13}$$

Changing the angular momentum of the flywheel to be represented in the n-frame

$$\vec{h} = (h_1 + h_2) \begin{bmatrix} \cos\gamma_2 * \cos\gamma_1 \\ \sin\gamma_2 \\ -\cos\gamma_2 * \sin\gamma_1 \end{bmatrix} \tag{A.14}$$

Changing the angular momentum of the flywheel to be represented in the n-frame

$$\dot{\vec{h}} = (h_1 + h_2) \begin{bmatrix} -\sin\gamma_1 \cos\gamma_2 & -\cos\gamma_1 \sin\gamma_2 \\ \sin\gamma_2 & \cos\gamma_2 \end{bmatrix} \tag{A.15}$$

Substituting the A.15, A.14, and A.3 a general equation is obtained to represent the EOM of the spacecraft with CMG.

$$\begin{bmatrix} I_x\dot{\omega}_x \\ I_y\dot{\omega}_y \\ I_z\dot{\omega}_z \end{bmatrix} = -\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \times \left(\begin{bmatrix} I_{xx}\omega_x \\ I_{yy}\omega_y \\ I_{zz}\omega_z \end{bmatrix} + h \begin{bmatrix} \cos\gamma_2\cos\gamma_1 \\ \sin\gamma_2 \\ -\cos\gamma_2\sin\gamma_1 \end{bmatrix} \right) - \\ ...h \begin{bmatrix} -\sin\gamma_1\cos\gamma_2 & -\cos\gamma_1\sin\gamma_2 \\ \sin\gamma_2 & \cos\gamma_2 \\ -\cos\gamma_1\cos\gamma_2 & \sin\gamma_1\sin\gamma_2 \end{bmatrix} \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix} + \vec{T}_{ext} \tag{A.16}$$

Simplifying

$$\begin{bmatrix} I_x\dot{\omega}_x \\ I_y\dot{\omega}_y \\ I_z\dot{\omega}_z \end{bmatrix} = \begin{bmatrix} (I_z - I_y)\,\omega_y\omega_z \\ (I_x - I_z)\,\omega_z\omega_x \\ (I_y - I_x)\,\omega_y\omega_x \end{bmatrix} - h \begin{bmatrix} -\sin\gamma_2\omega_z - \cos\gamma_2\sin\gamma_1\omega_y \\ \cos\gamma_2\cos\gamma_1\omega_z - \cos\gamma_2\sin\gamma_1\omega_x \\ -\cos\gamma_2\cos\gamma_1\omega_y + \sin\gamma_2\omega_x \end{bmatrix} - ... \\ ...h \begin{bmatrix} -\sin\gamma_1\cos\gamma_2 & -\cos\gamma_1\sin\gamma_2 \\ \sin\gamma_2 & \cos\gamma_2 \\ -\cos\gamma_1\cos\gamma_2 & \sin\gamma_1\sin\gamma_2 \end{bmatrix} \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix} + \vec{T}_{ext} \tag{A.17}$$

Substituting the gravity gradient equation for the T_{ext}

$$\vec{T}_{ext} + \begin{bmatrix} -3n^2\,(J_y - J_z)\,\phi \\ 3n^2\,(J_z - J_x)\,\theta \\ 0 \end{bmatrix} \tag{A.18}$$

$$\begin{bmatrix} I_x\dot{\omega}_x \\ I_y\dot{\omega}_y \\ I_z\dot{\omega}_z \end{bmatrix} = \begin{bmatrix} (I_y - I_z)\,\omega_y\omega_z + h\,(-\omega_z \sin\gamma_2 - \omega_y \cos\gamma_2 \sin\gamma_1) \\ (I_z - I_x)\,\omega_z\omega_x + h\,(\omega_z \cos\gamma_2 \cos\gamma_1 + \omega_y \cos\gamma_2 \sin\gamma_1) \\ (I_x - I_y)\,\omega_x\omega_y + h\,(\omega_x \sin\gamma_2 - \omega_y \cos\gamma_2 \cos\gamma_1) \end{bmatrix} - \ldots$$
$$\ldots h \begin{bmatrix} -\sin\gamma_1 \cos\gamma_2 & -\cos\gamma_1 \sin\gamma_2 \\ \sin\gamma_2 & \cos\gamma_2 \\ -\cos\gamma_1 \cos\gamma_2 & \sin\gamma_1 \sin\gamma_2 \end{bmatrix} \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix} + \begin{bmatrix} -3n^2\,(J_y - J_z)\,\phi \\ 3n^2\,(J_z - J_x)\,\theta \\ 0 \end{bmatrix} \tag{A.19}$$

Transforming the body angles into attitude angles

$$\begin{bmatrix} I_x\left(\ddot{\phi} - n\dot{\psi}\right) \\ I_y\ddot{\theta} \\ I_z\left(\ddot{\psi} + n\dot{\theta}_1\right) \end{bmatrix} = \begin{bmatrix} (I_y - I_z)\left(\dot{\theta}\dot{\psi} + \dot{\theta}\phi n - \dot{\psi} - n^2\phi\right) \\ (I_z - I_x)\left(\dot{\psi}\dot{\phi} - n\dot{\psi}\psi + n\phi\dot{\phi} - n^2\phi\psi\right) \\ (I_x - I - y)\left(\dot{\phi}\dot{\theta} - n\dot{\phi} - n\dot{\theta}\psi + n^2\psi\right) \end{bmatrix} + \ldots$$
$$\ldots h \begin{bmatrix} (-\dot{\psi} - n\phi)\sin\gamma_2 - \left(\dot{\theta} - n\right)\cos\gamma_2 \sin\gamma_1 \\ (\dot{\psi} + n\phi)\cos\gamma_2 \cos\gamma_1 + \left(\dot{\phi} - n\psi\right)\cos\gamma_2 \sin\gamma_1 \\ \left(\dot{\phi} - n\psi\right)\sin\gamma_2 - \left(\dot{\theta} - n\right)\cos\gamma_2 \cos\gamma_1 \end{bmatrix} - \ldots \tag{A.20}$$
$$\ldots h \begin{bmatrix} -\sin\gamma_1 \cos\gamma_2 & -\cos\gamma_1 \sin\gamma_2 \\ \sin\gamma_2 & \cos\gamma_2 \\ -\cos\gamma_1 \cos\gamma_2 & \sin\gamma_1 \sin\gamma_2 \end{bmatrix} \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix} + \begin{bmatrix} -3n^2\,(J_y - J_z)\,\phi \\ 3n^2\,(J_z - J_x)\,\theta \\ 0 \end{bmatrix}$$

Simplifying the equation

$$\begin{bmatrix} I_x\ddot{\phi} \\ I_y\ddot{\theta} \\ I_z\ddot{\psi} \end{bmatrix} = \begin{bmatrix} n\,(I_x - I_y + I_z)\,\dot{\psi} - 4n^2\,(I_y - I_z)\,\phi \\ 3n^2\,(I_z - I_x)\,\dot{\theta} \\ n\,(I_x + I_y - I_z)\,\dot{\phi} + n^2\,(I_x - I_y)\,\psi \end{bmatrix} + \ldots$$
$$\ldots h \begin{bmatrix} -\sin\gamma_2\dot{\psi} - \sin\gamma_2 n\phi - \dot{\theta}\cos\gamma_2 \sin\gamma_1 + n\cos\gamma_2 \sin\gamma_1 \\ \cos\gamma_2 \cos\gamma_1\dot{\psi} + \cos\gamma_2 \cos\gamma_1 n\phi + \cos\gamma_2 \sin\gamma_1\dot{\phi} - \cos\gamma_2 \sin\gamma_1 n\psi \\ \sin\gamma_2\dot{\phi} - \sin\gamma_2 n\psi - \cos\gamma_2 \cos\gamma_1\dot{\theta} + \cos\gamma_2 \cos\gamma_1 n \end{bmatrix} - \ldots \tag{A.21}$$
$$\ldots h \begin{bmatrix} -\sin\gamma_1 \cos\gamma_2 & -\cos\gamma_1 \sin\gamma_2 \\ \sin\gamma_2 & \cos\gamma_2 \\ -\cos\gamma_1 \cos\gamma_2 & \sin\gamma_1 \sin\gamma_2 \end{bmatrix} \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix}$$

Expanding and separating parts in terms of input signals and state-variables. A full EOM is obtained for the spacecraft with a DGCMG at LEO.

$$\begin{bmatrix} I_x\ddot{\phi} \\ I_y\ddot{\theta} \\ I_z\ddot{\psi} \end{bmatrix} =$$

$$\begin{bmatrix} \left[-4n^2\left(I_y-I_z\right)-hns_{\gamma_2}\right]\phi+h\left(-c_{\gamma_2}s_{\gamma_1}\right)\dot{\theta}+\left[n\left(I_x-I_y+I_z\right)-hs_{\gamma_2}\right]\dot{\psi}+nc_{\gamma_2}s_{\gamma_1} \\ \left[hnc_{\gamma_2}c_{\gamma_1}\right]\phi+\left[-hnc_{\gamma_2}s_{\gamma_1}\right]\psi+\left[hc_{\gamma_2}s_{\gamma_1}\right]\dot{\phi}+\left[3n^2\left(I_z-I_x\right)\right]\dot{\theta}+\left[h\cos\gamma_2\cos\gamma_1\right]\dot{\psi} \\ \left[n^2\left(I_x-I_y\right)-hns_{\gamma_2}\right]\psi+\left[n\left(-I_x+I_y-I_z\right)+hs_{\gamma_2}\right]\dot{\phi}+\left[hc_{\gamma_2}c_{\gamma_1}\right]\dot{\theta}+c_{\gamma_2}c_{\gamma_1}n \end{bmatrix} \ldots \quad \text{(A.22)}$$

$$\ldots - h\begin{bmatrix} -s_{\gamma_1}c_{\gamma_2} & -c_{\gamma_1}s_{\gamma_2} \\ s_{\gamma_2} & c_{\gamma_2} \\ -c_{\gamma_1}c_{\gamma_2} & s_{\gamma_1}s_{\gamma_2} \end{bmatrix}\begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix}$$

Assumptions are placed on the terms that aren't multiplied by the state variables. These terms are thought to be near-constant or considerably small, and they are assumed to be neglectable.

$$\begin{bmatrix} I_x\ddot{\phi} \\ I_y\ddot{\theta} \\ I_z\ddot{\psi} \end{bmatrix} =$$

$$\begin{bmatrix} \left[-4n^2\left(I_y-I_z\right)-hns_{\gamma_2}\right]\phi+h\left(-c_{\gamma_2}s_{\gamma_1}\right)\dot{\theta}+\left[n\left(I_x-I_y+I_z\right)-hs_{\gamma_2}\right]\dot{\psi} \\ \left[hnc_{\gamma_2}c_{\gamma_1}\right]\phi+\left[-hnc_{\gamma_2}s_{\gamma_1}\right]\psi+\left[hc_{\gamma_2}s_{\gamma_1}\right]\dot{\phi}+\left[3n^2\left(I_z-I_x\right)\right]\dot{\theta}+\left[h\cos\gamma_2\cos\gamma_1\right]\dot{\psi} \\ \left[n^2\left(I_x-I_y\right)-hns_{\gamma_2}\right]\psi+\left[n\left(-I_x+I_y-I_z\right)+hs_{\gamma_2}\right]\dot{\phi}+\left[hc_{\gamma_2}c_{\gamma_1}\right]\dot{\theta} \end{bmatrix} \ldots \quad \text{(A.23)}$$

$$\ldots - h\begin{bmatrix} -s_{\gamma_1}c_{\gamma_2} & -c_{\gamma_1}s_{\gamma_2} \\ s_{\gamma_2} & c_{\gamma_2} \\ -c_{\gamma_1}c_{\gamma_2} & s_{\gamma_1}s_{\gamma_2} \end{bmatrix}\begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix}$$

Once placed into a LPV state space, each index major element can be shown below. Refer to $\gamma_1 = \rho_1$ and $\gamma_2 = \rho_2$ when converting it to the LPV model in chapter 2.

$$
\begin{aligned}
a_{41} &= -4n^2\left(I_y - I_z\right) - hns_{\gamma_2} \\
a_{45} &= h(-c_{\gamma_2}s_{\gamma_1}) \\
a_{46} &= n\left(I_x - I_y + I_z\right) - hs_{\gamma_2} \\
a_{51} &= hnc_{\gamma_2}c_{\gamma_1} \\
a_{52} &= -hnc_{\gamma_2}s_{\gamma_1} \\
a_{53} &= hc_{\gamma_2}s_{\gamma_1} \\
a_{54} &= 3n^2\left(I_z - I_x\right) \\
a_{56} &= h\cos\gamma_2\cos\gamma_1 \\
a_{63} &= n^2\left(I_x - I_y\right) - hns_{\gamma_2} \\
a_{64} &= n\left(-I_x + I_y - I_z\right) + hs_{\gamma_2} \\
a_{65} &= hc_{\gamma_2}c_{\gamma_1}
\end{aligned}
\tag{A.24}
$$

Appendix B — MATLAB Code: Main Script for Testbed Simulation [State-Space, LQR Gain Values, and Simulation Models]

```
%% --------  [Double Gimbal Gyroscope for Testbed]  -------- %%
% MATLAB code used to run a LPV model and a gain-scheduling LQR
% controller for a control moment gyroscope testbed. Code also
% runs simulations that test the dynamics of the model. This
% included open-loop, closed-loop, and closed-loop tracking
% simulation.
% By: Jason Nguyen

clc; clear all; close all;

%% ------------------  [Simulation Time]  ------------------ %%

tf = 50; %Simulation Run-time

%% ----------------- [Testbed Parameters]  ----------------- %%


h1 = 0.005;       % flywheel angular momentum 1 [kg*m^2/s]
h2 = 0.005;       % flywheel angular momentum 2 [kg*m^2/s]
Iy =  0.02;       % moment of inertia about the y-axis
total_h = - (h1 + h2)/Iy;

% Gimbal Angles
gamma1_min = deg2rad(-90); % lower bound
gamma1_max = deg2rad(90); % upper bound
gamma1_initial = deg2rad(0); % initial condition for outer
   gimbal

gamma2_min = deg2rad(-90); % lower bound
gamma2_max = deg2rad(90); % upper bound
gamma2_initial = deg2rad(0); % initial condition for inner
   gimbal

%% -----------------  [LQR Weight Matrix]  ----------------- %%

% State Weight Matrix
Q = [50,0;
    0,5];
```

```
37
38 % Actuator Weight Matrix
39 R = [10,0;
40     0,10];
41
42 %% -------  [Defining LPV Parameters & State-Space]  ------- %%
43
44 fh = @(t,p) cmg(t,p,total_h,gamma1_min,gamma1_max,gamma2_min,
       gamma2_max,Iy);
45 G = lpvss(["p1","p2"],fh);
46
47 p1 = linspace(gamma1_min,gamma1_max,10); % setting intervals
       for LPV model
48 p2 = linspace(gamma2_min,gamma2_max,10); % setting intervals
       for LPV model
49
50 [p1_grid,p2_grid] = ndgrid(p1,p2);
51
52 ICs = sample(G,[],p1_grid,p2_grid);
53
54 %% -------------  [LPV State-Space Equation]  ------------- %%
55
56 function [A,B,C,D,E,dx0,x0,u0,y0,Delays] = cmg(t,p,AM,
       gamma1_min,gamma1_max,gamma2_min,gamma2_max,Iy)
57
58     p(1) = max(gamma1_min,min(p(1),gamma1_max));
59     p(2) = max(gamma2_min,min(p(2),gamma2_max));
60
61     A = [0 1 ; 0 0/Iy];
62     B = [0 0 ; AM*sin(p(2)) AM*cos(p(2))];
63     C = [1 0 ; 0 1];
64     D = [0 0; 0 0];
65     E = [];
66
67     % no offsets or delays
68     dx0 = []; x0 = []; u0 = []; y0 = []; Delays = [];
69
70 end
71
72 %% ----------  [Defining Controller LQR Gains]  ---------- %%
73
74 length(ICs);
75 for i = 1:length(ICs)
76 for l = 1:length(ICs)
77 K_gains{i,l} = lqr(ICs(:,:,i,l),Q,R);
```

```
78 K1(i,l) = K_gains{i,l}(1,1);
79 K2(i,l) = K_gains{i,l}(1,2);
80 K3(i,l) = K_gains{i,l}(2,1);
81 K4(i,l) = K_gains{i,l}(2,2);
82 end
83 end
84
85 %% --------------------  [Simulink]  -------------------- %%
86 % Note: Comment out function to remove preferred simulations
87
88 CMG_Testbed_Attitude_Openloop();
89
90 CMG_Testbed_Attitude_Closedloop();
91
92 CMG_Testbed_Attitude_ClosedloopTracking();
```

Appendix C — MATLAB Code: Subscript for Testbed Simulation [Open-Loop]

```
%% --------------------  [Simulink]  -------------------- %%

open_system('Testbed_OpenLoop');
OL = sim('Testbed_OpenLoop');

%% ----------------------  [Plot]  ---------------------- %%

figure;
hold on;
subplot(2,1,1);
plot(OL.Attitude_Angles.time(:),OL.Attitude_Angles.signals.
   values(:,1),"Color","#0072BD");
legend('Yaw');
title('Open-Loop Response [Yaw]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
grid on;

subplot(2,1,2);
plot(OL.Attitude_Angles_Rates.time(:),OL.Attitude_Angles_Rates.
   signals.values(:,1),"Color","#D95319");
legend('Yaw Rate');
title('Open-Loop Response [Yaw Rate]');
xlabel('Time [Seconds]');
ylabel('Angular [Degrees/s]');
grid on;

figure;
hold on;
subplot(2,2,1);
plot(OL.gamma_output.time(:),OL.gamma_output.signals.values
   (:,1),"Color","#0072BD");
legend('Outer Gimbal');
title('Open-Loop Response [Outer Gimbal]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.gamma_output.time(1) OL.gamma_output.time(length(OL.
   gamma_output.time(:)))]);
grid on;

```

```
subplot(2,2,2);
plot(OL.gamma_output.time(:),OL.gamma_output.signals.values
   (:,2),"Color","#0072BD");
legend('Inner Gimbal');
title('Open-Loop Response [Inner Gimbal]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.gamma_output.time(1) OL.gamma_output.time(length(OL.
   gamma_output.time(:)))]);
grid on;

subplot(2,2,3);
plot(OL.gamma_output_rates.time(:),OL.gamma_output_rates.
   signals.values(:,1),"Color","#D95319");
legend('Outer Gimbal');
title('Open-Loop Response [Outer Gimbal Rate]');
xlabel('Time [Seconds]');
ylabel('Angle Rate[Degrees/s]');
xlim([OL.gamma_output_rates.time(1) OL.gamma_output_rates.time(
   length(OL.gamma_output_rates.time(:)))]);
grid on;

subplot(2,2,4);
plot(OL.gamma_output_rates.time(:),OL.gamma_output_rates.
   signals.values(:,2),"Color","#D95319");
legend('Inner Gimbal');
title('Open-Loop Response [Inner Gimbal Rate]');
xlabel('Time [Seconds]');
ylabel('Angle Rate[Degrees/s]');
xlim([OL.gamma_output_rates.time(1) OL.gamma_output_rates.time(
   length(OL.gamma_output_rates.time(:)))]);
grid on;

figure;
hold on;
Gpole = pole(ICs);
Gpole = Gpole(:);
plot(real(Gpole),imag(Gpole),'bx');
title('Eigenvalue');
xlabel('Real');
ylabel('Imag')
grid on;
pole = eig(ICs(:,:,1,1))
```

Appendix D — MATLAB Code: Subscript for Testbed Simulation [Closed-Loop]

```
%% --------------------  [Simulink]  -------------------- %%

open_system('Testbed_ClosedLoop');
OL = sim('Testbed_ClosedLoop');

%% ----------------------  [Plot]  ---------------------- %%

figure;
hold on;
subplot(2,1,1);
plot(OL.Attitude_Angles.time(:),OL.Attitude_Angles.signals.
   values(:,1),"Color","#0072BD");
legend('Yaw');
title('LQR Response [Yaw]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
%xlim([OL.Attitude_Angles.time(1) OL.Attitude_Angles.time(:)]);
grid on;

% Yaw Rate
subplot(2,1,2);
plot(OL.Attitude_Angles_Rates.time(:),OL.Attitude_Angles_Rates.
   signals.values(:,1),"Color","#D95319");
legend('Yaw Rate');
title('LQR Response [Yaw Rate]');
xlabel('Time [Seconds]');
ylabel('Angular [Degrees/s]');
%xlim([OL.Attitude_Angles_Rates.time(1) 50000]);
grid on;

% Gimbal Results
figure;
hold on;
subplot(2,2,1);
plot(OL.gamma_output.time(:),OL.gamma_output.signals.values
   (:,1),"Color","#0072BD");
legend('Outer Gimbal');
title('LQR Response [Outer Gimbal]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
```

```
xlim([OL.gamma_output.time(1) OL.gamma_output.time(length(OL.
    gamma_output.time(:)))]);
grid on;

subplot(2,2,2);
plot(OL.gamma_output.time(:),OL.gamma_output.signals.values
    (:,2),"Color","#0072BD");
legend('Inner Gimbal');
title('LQR Response [Inner Gimbal]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.gamma_output.time(1) OL.gamma_output.time(length(OL.
    gamma_output.time(:)))]);
grid on;

subplot(2,2,3);
hold on;
plot(OL.gamma_output_rates.time(:),OL.gamma_output_rates.
    signals.values(:,1),"Color","#D95319");
plot(OL.Input.time(:),OL.Input.signals.values(:,1),"Color","#77
    AC30",'LineStyle','--');
legend('Outer Gimbal','Input Outer Gimbal');
title('LQR Response [Outer Gimbal Rate]');
xlabel('Time [Seconds]');
ylabel('Angle Rate[Degrees/s]');
xlim([OL.gamma_output_rates.time(1) OL.gamma_output_rates.time(
    length(OL.gamma_output_rates.time(:)))]);
grid on;

subplot(2,2,4);
hold on;
plot(OL.gamma_output_rates.time(:),OL.gamma_output_rates.
    signals.values(:,2),"Color","#D95319");
plot(OL.Input.time(:),OL.Input.signals.values(:,2),"Color","#77
    AC30",'LineStyle','--');
legend('Inner Gimbal','Input Outer Gimbal');
title('LQR Response [Inner Gimbal Rate]');
xlabel('Time [Seconds]');
ylabel('Angle Rate [Deg/s]');
xlim([OL.gamma_output_rates.time(1) OL.gamma_output_rates.time(
    length(OL.gamma_output_rates.time(:)))]);
grid on;
```

Appendix E — MATLAB Code: Subscript for Testbed Simulation [Closed-Loop LQR-Tracking]

```
%% --------------------  [Simulink]  -------------------- %%

open_system('Testbed_ClosedLoopTracking');
OL = sim('Testbed_ClosedLoopTracking');

%% ----------------------  [Plot]  ---------------------- %%

figure;
subplot(2,1,1);
hold on;
plot(OL.Attitude_Angles.time(:),OL.Attitude_Angles.signals.
   values(:,1),"Color","#0072BD");
plot(OL.Input.time(:),OL.Input.signals.values(:,1),"Color","#77
   AC30",'LineStyle','--');
legend('Yaw','Desired Yaw');
title('LQR Track Response [Yaw]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
%xlim([OL.Attitude_Angles.time(1) OL.Attitude_Angles.time(:)]);
grid on;

% Yaw Rate
subplot(2,1,2);
plot(OL.Attitude_Angles_Rates.time(:),OL.Attitude_Angles_Rates.
   signals.values(:,1),"Color","#D95319");
legend('Yaw Rate');
title('LQR Track Response [Yaw Rate]');
xlabel('Time [Seconds]');
ylabel('Angular [Degrees/s]');
%xlim([OL.Attitude_Angles_Rates.time(1) 50000]);
grid on;

% Gimbal Results
figure;
hold on;
subplot(2,2,1);
plot(OL.gamma_output.time(:),OL.gamma_output.signals.values
   (:,1),"Color","#0072BD");
legend('Outer Gimbal');
title('LQR Track Response [Outer Gimbal]');
```

```
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.gamma_output.time(1) OL.gamma_output.time(length(OL.
    gamma_output.time(:)))]);
grid on;

subplot(2,2,2);
plot(OL.gamma_output.time(:),OL.gamma_output.signals.values
    (:,2),"Color","#0072BD");
legend('Inner Gimbal');
title('LQR Track Response [Inner Gimbal]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.gamma_output.time(1) OL.gamma_output.time(length(OL.
    gamma_output.time(:)))]);
grid on;

subplot(2,2,3);
plot(OL.gamma_output_rates.time(:),OL.gamma_output_rates.
    signals.values(:,1),"Color","#D95319");
legend('Outer Gimbal');
title('LQR Track [Outer Gimbal Rate]');
xlabel('Time [Seconds]');
ylabel('Angle Rate[Deg/s]');
xlim([OL.gamma_output_rates.time(1) OL.gamma_output_rates.time(
    length(OL.gamma_output_rates.time(:)))]);
grid on;

subplot(2,2,4);
plot(OL.gamma_output_rates.time(:),OL.gamma_output_rates.
    signals.values(:,2),"Color","#D95319");
legend('Inner Gimbal');
title('LQR Track [Inner Gimbal Rate]');
xlabel('Time [Seconds]');
ylabel('Angle Rate [Deg/s]');
xlim([OL.gamma_output_rates.time(1) OL.gamma_output_rates.time(
    length(OL.gamma_output_rates.time(:)))]);
grid on;
```

Appendix F — MATLAB Code: Main Script for Spacecraft Simulation [State-Space, LQR Gain Values, and Simulation Models]

```
%% ------  [Double Gimbal Gyroscope for Satellite]  ------ %%
clc; clear all; close all;
% By: Jason Nguyen

%% --------------  [Spacecraft Parameters]  -------------- %%

% Angular Momentum of Flywheels
h1 = 0.0100;
h2 = 0.0100;

%MIST <https://www.mdpi.com/2226-4310/10/4/378>
Ix = 0.037;
Iy = 0.051;
Iz = 0.021;

%sSAMsat <https://www.mdpi.com/2226-4310/10/4/378>
%Ix = 0.00402;
%Iy = 0.01422;
%Iz = 0.01454;

%MIST <Attitude Control of a 3U CubeSat - Iraqi Satellite (
   TIGRISAT)>
%Ix = 0.0333;
%Iy = 0.0333;
%Iz = 6.667*10^-3;

% Gimbal Angles
gamma1_min = deg2rad(-90);
gamma1_max = deg2rad(90);
gamma1_initial = deg2rad(0);
gamma1_points = 16+1;

gamma2_min = deg2rad(0);
gamma2_max = deg2rad(180);
gamma2_initial = deg2rad(90);
gamma2_points = 16+1;

%% --------  [Low Earth Orbit (LEO) Conditions]  -------- %%
```

```
38
39 R_earth = 6371*10^3;     % Radius of Earth
40 a = 600*10^3;            % Satellite Altitude
41 r = R_earth + a;         % Radius of Satellite to Center of
      Earth
42 u = 3.986*10^14;         % Earth's gravitational parameter
43 n = sqrt(u/(r^3));       % Angular velocity
44
45 %% ----------------  [LQR Weight Matrix]  ---------------- %%
46
47 % [Better Capablility in Tracking with the LQR Tracking (Can do
       Roll and Pitch)] %
48  Q = [2,0,0,0,0,0;...
49       0,2,0,0,0,0;...
50       0,0,.001,0,0,0;...
51       0,0,0,1,0,0
52       0,0,0,0,1,0
53       0,0,0,0,0,1];
54
55 % Actuator Weight Matrix
56 R = [1,0;
57      0,1];
58
59 %% -----  [Defining LPV Parameters & State-Space]  ----- %%
60
61 p1 = linspace(gamma1_min,gamma1_max,gamma1_points);  % Outter
      Gimbal
62 p2 = linspace(gamma2_min,gamma2_max,gamma2_points); % Inner
      Gimbal
63
64 fh = @(t,p) cmg(t,p,Ix,Iy,Iz,n,h1,h2,gamma1_min,gamma1_max,
      gamma2_min,gamma2_max);       % Calls Statespace Function
65 G = lpvss(["p1","p2"],fh);                    % LPV Statespace
66
67 [p1_grid,p2_grid] = ndgrid(p1,p2);
68 ICs = sample(G,[],p1_grid,p2_grid);          % List of
      Statespace
69
70 %% ---------------- [Controllability]  ---------------- %%
71
72 Co = ctrb(ICs(:,:,1,1));
73 hi(1,:) = rank(Co)
74
75 bode(ICs(:,:,1,1));
76
```

```
%% ---------  [Defining Controller LQR Gains]  --------- %%

for i = 1:length(ICs)
    for l = 1:length(ICs)
        [K_gains{i,l},s,P{i,1}] = lqr(ICs(:,:,i,l),Q,R);
        K11(i,l) = K_gains{i,l}(1,1);
        K21(i,l) = K_gains{i,l}(2,1);
        K12(i,l) = K_gains{i,l}(1,2);
        K22(i,l) = K_gains{i,l}(2,2);
        K13(i,l) = K_gains{i,l}(1,3);
        K23(i,l) = K_gains{i,l}(2,3);
        K14(i,l) = K_gains{i,l}(1,4);
        K24(i,l) = K_gains{i,l}(2,4);
        K15(i,l) = K_gains{i,l}(1,5);
        K25(i,l) = K_gains{i,l}(2,5);
        K16(i,l) = K_gains{i,l}(1,6);
        K26(i,l) = K_gains{i,l}(2,6);
    end
end

%% ---------------------  [Poles]  --------------------- %%

Gpole = pole(ICs);
Gpole = Gpole(:);

plot(real(Gpole),imag(Gpole),'bx');
title('Pole-Zero Map');
xlabel('Real');
ylabel('Imag')
grid on;

%% ----  [Defining Dynamical Model for State-Space]  ---- %%

function [A,B,C,D,E,dx0,x0,u0,y0,Delays] = cmg(t,p,Ix,Iy,Iz,n,
   h1,h2,gamma1_min,gamma1_max,gamma2_min,gamma2_max)

    p(1) = max(gamma1_min,min(p(1),gamma1_max));
    p(2) = max(gamma2_min,min(p(2),gamma2_max));

    A = [0 0 0 1 0 0
        0 0 0 0 1 0
        0 0 0 0 0 1
        (-4*(n^2)*(Iy-Iz)-(h1+h2)*n*sin(p(2)))/Ix 0 0 0 (-(h1+
           h2)*cos(p(2))*sin(p(1)))/Ix (n*(Ix-Iy+Iz)-(h1+h2)*
           sin(p(2)))/Ix
```

```
        (n*(h1+h2)*cos(p(2))*cos(p(1)))/Iy (3*(n^2)*(Iz-Ix))/Iy
            (-n*(h1+h2)*cos(p(2))*sin(p(1)))/Iy ((h1+h2)*cos(p
           (2))*sin(p(1)))/Iy 0 ((h1+h2)*cos(p(2))*cos(p(1)))/
           Iy
        0 0 ((n^2)*(Ix-Iy)-n*(h1+h2)*sin(p(2)))/Iz (n*(-Ix+Iy-
           Iz)+(h1+h2)*sin(p(2)))/Iz ((h1+h2)*cos(p(2))*cos(p
           (1)))/Iz 0];

    B = -(h1+h2)*[0 0
        0 0
        0 0
        -sin(p(1))*cos(p(2))/Ix -cos(p(1))*sin(p(2))/Ix
        sin(p(2))/Iy cos(p(2))/Iy
        -cos(p(1))*cos(p(2))/Iz sin(p(1))*sin(p(2))/Iz];

    C = [1 0 0 0 0 0
        0 1 0 0 0 0
        0 0 1 0 0 0
        0 0 0 1 0 0
        0 0 0 0 1 0
        0 0 0 0 0 1];

    D = [0 0
        0 0
        0 0
        0 0
        0 0
        0 0];

    E = [];
    % no offsets or delays
    dx0 = []; x0 = []; u0 = []; y0 = []; Delays = [];

end

Q1 = [1,0,0,0,0,0,0,0,0,0,0,0;...
      0,1,0,0,0,0,0,0,0,0,0,0;...
      0,0,1,0,0,0,0,0,0,0,0,0;...
      0,0,0,1,0,0,0,0,0,0,0,0;
      0,0,0,0,1,0,0,0,0,0,0,0;
      0,0,0,0,0,1,0,0,0,0,0,0;
      0,0,0,0,0,0,1,0,0,0,0,0;
      0,0,0,0,0,0,0,1,0,0,0,0;
      0,0,0,0,0,0,0,0,1,0,0,0;
      0,0,0,0,0,0,0,0,0,1,0,0;
```

```
      0,0,0,0,0,0,0,0,0,0,1,0;
      0,0,0,0,0,0,0,0,0,0,0,1;];

R = [1,0;
    0,1];

%% ------------  [Determine Controllability]  ------------ %%

for i = 1:length(ICs)
    for l = 1:length(ICs)
        Co_rank(i,l) = rank(ctrb(ICs(:,:,i,l)));
    end
end

%% --------------------  [Simulink]  -------------------- %%

CMG_Spacecraft_Attitude_Control_Openloop();

CMG_Spacecraft_Attitude_Control_LQR();

CMG_Spacecraft_Attitude_Control_LQRTracking();
```

Appendix G — MATLAB Code: Subscript for Spacecraft Simulation [Open-Loop]

```
%% --------------------  [Simulink]  -------------------- %%

tf = 500000;
%tf = 50000000;

open_system('LPVattitudeOpenLoop');
OL = sim('LPVattitudeOpenLoop');

%% ----------------------  [Plot]  ---------------------- %%

% Roll
figure;

hold on;
subplot(3,1,1);
plot(OL.Attitude_Angles.time(:),OL.Attitude_Angles.signals.
   values(:,1),"Color","#0072BD");
legend('Roll');
title('Open-Loop Response [Roll]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.Attitude_Angles.time(1) 30000]);
grid on;

% Pitch
subplot(3,1,2);
plot(OL.Attitude_Angles.time(:),OL.Attitude_Angles.signals.
   values(:,2),"Color","#D95319");
legend('Pitch');
title('Open-Loop Response [Pitch]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.Attitude_Angles.time(1) 500000]);
grid on;

% Yaw
subplot(3,1,3);
plot(OL.Attitude_Angles.time(:),OL.Attitude_Angles.signals.
   values(:,3),"Color","#7E2F8E");
legend('Yaw');
```

```
38 title('Open-Loop Response [Yaw]');
39 xlabel('Time [Seconds]');
40 ylabel('Angle [Degrees]');
41 xlim([OL.Attitude_Angles.time(1) 50000]);
42 grid on;
43
44 % Yaw Rate
45 figure;
46 hold on;
47 subplot(3,1,1);
48 plot(OL.Attitude_Angles_Rates.time(:),OL.Attitude_Angles_Rates.
      signals.values(:,1),"Color","#0072BD");
49 legend('Roll Rate');
50 title('Open-Loop Response [Roll Rate]');
51 xlabel('Time [Seconds]');
52 ylabel('Angular [Deg/s]');
53 xlim([OL.Attitude_Angles_Rates.time(1) 30000]);
54 grid on;
55
56 % Pitch Rate
57 subplot(3,1,2);
58 plot(OL.Attitude_Angles_Rates.time(:),OL.Attitude_Angles_Rates.
      signals.values(:,2),"Color","#D95319");
59 legend('Pitch Rate');
60 title('Open-Loop Response [Pitch Rate]');
61 xlabel('Time [Seconds]');
62 ylabel('Angular [Deg/s]');
63 xlim([OL.Attitude_Angles_Rates.time(1) 500000]);
64 grid on;
65
66 % Roll Rate
67 subplot(3,1,3);
68 plot(OL.Attitude_Angles_Rates.time(:),OL.Attitude_Angles_Rates.
      signals.values(:,3),"Color","#7E2F8E");
69 legend('Yaw Rate');
70 title('Open-Loop Response [Yaw Rate]');
71 xlabel('Time [Seconds]');
72 ylabel('Angular [Deg/s]');
73 xlim([OL.Attitude_Angles_Rates.time(1) 50000]);
74 grid on;
75
76 % Gimbal Results
77 figure;
78 hold on;
79 subplot(2,2,1);
```

```
plot(OL.gamma_output.time(:),OL.gamma_output.signals.values
    (:,1),"Color","#0072BD");
legend('Outer Gimbal');
title('Open-Loop Response [Outer Gimbal]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.gamma_output.time(1) OL.gamma_output.time(length(OL.
    gamma_output.time(:)))]);
grid on;

subplot(2,2,2);
plot(OL.gamma_output.time(:),OL.gamma_output.signals.values
    (:,2),"Color","#0072BD");
legend('Inner Gimbal');
title('Open-Loop Response [Inner Gimbal]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.gamma_output.time(1) OL.gamma_output.time(length(OL.
    gamma_output.time(:)))]);
grid on;

subplot(2,2,3);
plot(OL.gamma_output_rates.time(:),OL.gamma_output_rates.
    signals.values(:,1),"Color","#D95319");
legend('Outer Gimbal');
title('Open-Loop Response [Outer Gimbal]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.gamma_output_rates.time(1) OL.gamma_output_rates.time(
    length(OL.gamma_output_rates.time(:)))]);
grid on;

subplot(2,2,4);
plot(OL.gamma_output_rates.time(:),OL.gamma_output_rates.
    signals.values(:,2),"Color","#D95319");
legend('Inner Gimbal');
title('Open-Loop Response [Inner Gimbal]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.gamma_output_rates.time(1) OL.gamma_output_rates.time(
    length(OL.gamma_output_rates.time(:)))]);
grid on;
```

Appendix H — MATLAB Code: Subscript for Spacecraft Simulation [Closed-Loop]

```
% By: Jason Nguyen
%% ---------------------  [Simulink]  --------------------- %%

tf = 30000;
open_system('LPVattitudeLQR');
OL = sim('LPVattitudeLQR');

%% ------------------------  [Plot]  ------------------------ %%

% Yaw
figure;
hold on;
subplot(3,1,1);
plot(OL.Attitude_Angles.time(:),OL.Attitude_Angles.signals.
   values(:,1),"Color","#0072BD");
legend('Roll');
title('LQR Response [Roll]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.Attitude_Angles.time(1) 100]);
ylim([min(OL.Attitude_Angles.signals.values(:,1))-5 max(OL.
   Attitude_Angles.signals.values(:,1))+5]);

grid on;

% Pitch
subplot(3,1,2);
plot(OL.Attitude_Angles.time(:),OL.Attitude_Angles.signals.
   values(:,2),"Color","#D95319");
legend('Pitch');
title('LQR Response [Pitch]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.Attitude_Angles.time(1) 100]);
ylim([min(OL.Attitude_Angles.signals.values(:,2))-5 max(OL.
   Attitude_Angles.signals.values(:,2))+5]);

grid on;

% Roll
```

```
subplot(3,1,3);
plot(OL.Attitude_Angles.time(:),OL.Attitude_Angles.signals.
   values(:,3),"Color","#7E2F8E");
legend('Yaw');
title('LQR Response [Yaw]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.Attitude_Angles.time(1) 30000]);
ylim([min(OL.Attitude_Angles.signals.values(:,3))-5 max(OL.
   Attitude_Angles.signals.values(:,3))+5]);

grid on;

% Yaw Rate
figure;
hold on;
subplot(3,1,1);
plot(OL.Attitude_Angles_Rates.time(:),OL.Attitude_Angles_Rates.
   signals.values(:,1),"Color","#0072BD");
legend('Roll Rate');
title('LQR Response [Roll Rate]');
xlabel('Time [Seconds]');
ylabel('Angular [Deg/s]');
xlim([OL.Attitude_Angles_Rates.time(1) 100]);
ylim([min(OL.Attitude_Angles_Rates.signals.values(:,1))-0.1 max
   (OL.Attitude_Angles_Rates.signals.values(:,1)+0.1)]);
grid on;

% Pitch Rate
subplot(3,1,2);
plot(OL.Attitude_Angles_Rates.time(:),OL.Attitude_Angles_Rates.
   signals.values(:,2),"Color","#D95319");
legend('Pitch Rate');
title('LQR Response [Pitch Rate]');
xlabel('Time [Seconds]');
ylabel('Angular [Deg/s]');
xlim([OL.Attitude_Angles_Rates.time(1) 100]);
ylim([min(OL.Attitude_Angles_Rates.signals.values(:,2))-0.1 max
   (OL.Attitude_Angles_Rates.signals.values(:,2)+0.1)]);

grid on;

% Roll Rate
subplot(3,1,3);
```

```
plot(OL.Attitude_Angles_Rates.time(:),OL.Attitude_Angles_Rates.
    signals.values(:,3),"Color","#7E2F8E");
legend('Yaw Rate');
title('LQR Response [Yaw Rate]');
xlabel('Time [Seconds]');
ylabel('Angular [Deg/s]');
xlim([OL.Attitude_Angles_Rates.time(1) 100]);
ylim([min(OL.Attitude_Angles_Rates.signals.values(:,3)) max(OL.
    Attitude_Angles_Rates.signals.values(:,3))]);

grid on;

% Gimbal Results
figure;
hold on;
subplot(2,2,1);
plot(OL.gamma_output.time(:),OL.gamma_output.signals.values
    (:,1),"Color","#0072BD");
legend('Outer Gimbal');
title('LQR Response [Outer Gimbal]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.gamma_output.time(1) 50]);
ylim([min(OL.gamma_output.signals.values(:,1))-20 max(OL.
    gamma_output.signals.values(:,1))+20]);

grid on;

subplot(2,2,2);
plot(OL.gamma_output.time(:),OL.gamma_output.signals.values
    (:,2),"Color","#0072BD");
legend('Inner Gimbal');
title('LQR Response [Inner Gimbal]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
xlim([OL.gamma_output.time(1) OL.gamma_output.time(length(OL.
    gamma_output.time(:)))]);
ylim([min(OL.gamma_output.signals.values(:,2))-20 max(OL.
    gamma_output.signals.values(:,2))+20]);

grid on;

subplot(2,2,3);
plot(OL.gamma_output_rates.time(:),OL.gamma_output_rates.
    signals.values(:,1),"Color","#D95319");
```

```
112 legend('Outer Gimbal');
113 title('LQR Response [Outer Gimbal]');
114 xlabel('Time [Seconds]');
115 ylabel('Angle [Deg/s]');
116 xlim([OL.gamma_output_rates.time(1) 50]);
117 grid on;
118
119 subplot(2,2,4);
120 plot(OL.gamma_output_rates.time(:),OL.gamma_output_rates.
       signals.values(:,2),"Color","#D95319");
121 legend('Inner Gimbal');
122 title('LQR Response [Inner Gimbal]');
123 xlabel('Time [Seconds]');
124 ylabel('Angle [Deg/s]');
125 xlim([OL.gamma_output_rates.time(1) 50]);
126 grid on;
```

Appendix I — MATLAB Code: Subscript for Spacecraft Simulation [Closed-Loop LQR-Tracking]

```
close all;
%% ---------------------  [Simulink]  --------------------- %%

tf = 10000;

%tf = 50000000;
open_system('LPVattitudeLQRTracking');
OL = sim('LPVattitudeLQRTracking');

%% ----------------------  [Plot]  ---------------------- %%

% Yaw
figure;
subplot(3,1,1);
hold on;
plot(OL.Attitude_Angles.time(:),OL.Attitude_Angles.signals.
   values(:,1),"Color","#0072BD");
legend('Roll');
title('LQR Response [Roll]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
grid on;

% Pitch
subplot(3,1,2);
hold on;
plot(OL.Attitude_Angles.time(:),OL.Attitude_Angles.signals.
   values(:,2),"Color","#D95319");
plot(OL.Input.time(:),OL.Input.signals.values(:,2),"Color","#77
   AC30",'LineStyle','--');
legend('Pitch','Input');
title('LQR Response [Pitch]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
grid on;

% Roll
subplot(3,1,3);
hold on;
```

```
plot(OL.Attitude_Angles.time(:),OL.Attitude_Angles.signals.
    values(:,3),"Color","#7E2F8E");
legend('Yaw','Input');
title('LQR Response [Yaw]');
xlabel('Time [Seconds]');
ylabel('Angle [Degrees]');
grid on;

% Yaw Rate
figure;
hold on;
subplot(3,1,1);
plot(OL.Attitude_Angles_Rates.time(:),OL.Attitude_Angles_Rates.
    signals.values(:,1),"Color","#0072BD");
legend('Roll Rate');
title('LQR Response [Roll Rate]');
xlabel('Time [Seconds]');
ylabel('Angular [Deg/s]');
grid on;

% Pitch Rate
subplot(3,1,2);
plot(OL.Attitude_Angles_Rates.time(:),OL.Attitude_Angles_Rates.
    signals.values(:,2),"Color","#D95319");
legend('Pitch Rate');
title('LQR Response [Pitch Rate]');
xlabel('Time [Seconds]');
ylabel('Angular [Deg/s]');
grid on;

% Roll Rate
subplot(3,1,3);
plot(OL.Attitude_Angles_Rates.time(:),OL.Attitude_Angles_Rates.
    signals.values(:,3),"Color","#7E2F8E");
legend('Yaw Rate');
title('LQR Response [Yaw Rate]');
xlabel('Time [Seconds]');
ylabel('Angular [Deg/s]');
grid on;

% Gimbal Results
figure;
hold on;
subplot(2,2,1);
```

```
77 plot(OL.gamma_output.time(:),OL.gamma_output.signals.values
      (:,1),"Color","#0072BD");
78 legend('Outer Gimbal');
79 title('LQR Response [Outer Gimbal]');
80 xlabel('Time [Seconds]');
81 ylabel('Angle [Degrees]');
82 grid on;
83
84 subplot(2,2,2);
85 plot(OL.gamma_output.time(:),OL.gamma_output.signals.values
      (:,2),"Color","#0072BD");
86 legend('Inner Gimbal');
87 title('LQR Response [Inner Gimbal]');
88 xlabel('Time [Seconds]');
89 ylabel('Angle [Degrees]');
90
91 grid on;
92
93 subplot(2,2,3);
94 plot(OL.gamma_output_rates.time(:),OL.gamma_output_rates.
      signals.values(:,1),"Color","#D95319");
95 legend('Outer Gimbal');
96 title('LQR Response [Outer Gimbal]');
97 xlabel('Time [Seconds]');
98 ylabel('Angle [Deg/s]');
99 grid on;
100
101 subplot(2,2,4);
102 plot(OL.gamma_output_rates.time(:),OL.gamma_output_rates.
      signals.values(:,2),"Color","#D95319");
103 legend('Inner Gimbal');
104 title('LQR Response [Inner Gimbal]');
105 xlabel('Time [Seconds]');
106 ylabel('Angle [Deg/s]');
107 grid on;
```

Appendix J — MATLAB Code: IMU Raw Data and Calibration for Accelerometer Scatterplot

```
%% -------  [Calibration for Accelerometer Scaling]  ------- %%
clc; clear all; close all;
% By: Jason Nguyen

%% --------------  [Raw Accelerometer Data]  -------------- %%

C = [-0.02153320313     -0.07476806641  10.06856689
-0.03349609375  -0.11484375     10.07574463
-0.1680786133   -3.692944336    9.424963379
0.05144042969   -8.794519043    5.054321289
0.057421875     -8.634814453    5.083630371
0.005981445313  -9.777868652    2.402746582
0.08314208984   -10.00157471    0.2165283203
0.06998291016   -10.09428711    0.3295776367
-0.09031982422  -9.477600098    -2.970385742
0.07596435547   -9.037963867    -4.063793945
0.03708496094   -8.162878418    -5.504724121
0.1794433594    -8.210131836    -5.449694824
-0.01076660156  -5.860620117    -7.772290039
0.1949951172    -3.610998535    -8.972766113
0.2464355469    -3.648681641    -8.966186523
0.1369750977    -0.3762329102   -9.611584473
-0.2183227539   -3.617578125    -8.991308594
-0.7105957031   -5.601623535    -7.929003906
-0.7883544922   -6.98692627     -6.837390137
-0.9923217773   -8.39017334     -5.037573242
-0.9815551758   -8.432641602    -5.035778809
-1.000695801    -8.803491211    -4.426867676
-0.3774291992   1.545007324     -9.433337402
-0.2787353516   1.640112305     -9.572106934
-0.2583984375   4.011755371     -8.458361816
-0.3487182617   5.296569824     -7.868591309
-0.1124511719   6.488671875     -6.977355957
-0.06220703125  6.508410645     -6.983935547
-0.5150024414   6.045446777     -7.107751465
0.07775878906   7.949938965     -5.123706055
0.09271240234   7.857226563     -5.462854004
0.4414306641    8.768200684     -3.563745117
0.4551879883    9.332250977     -1.829724121
0.4689453125    9.393859863     -1.795629883
```

```
0.5006469727    9.343615723     -1.77409668
0.1537231445    9.616369629     -0.363671875
0.09151611328   8.883642578     4.048242188
0.1405639648    8.843566895     4.083532715
0.1100585938    7.507312012     6.370837402
-0.019140625    4.94486084      8.513989258
-0.04844970703  4.83001709      8.685656738
-0.09151611328  2.464953613     9.622949219
0.005981445313  4.541113281     8.838183594
-0.07058105469  2.250219727     9.839477539
0.07177734375   0.01794433594   10.04224854
0.05981445313   0.03947753906   10.04703369
0.06998291016   0.03529052734   10.06138916
-2.224499512    -0.2201171875   9.640893555
-2.225695801    -0.2685668945   9.904077148
-2.273547363    -0.3403442383   9.971069336
-5.306140137    -0.09450683594  8.612084961
-5.359973145    -0.09749755859  8.481091309
-0.8344116211   -0.2721557617   10.06557617
-2.235266113    -0.258996582    9.822729492
-4.601525879    -0.1052734375   8.935681152
-6.440820313    -0.02093505859  7.652062988
-6.425268555    -0.03469238281  7.683764648
-6.405529785    -0.01256103516  7.66940918
-8.481689453    -0.05802001953  5.212231445
-9.717456055    -0.07297363281  1.756152344
-9.746765137    -0.07416992188  1.726245117
-9.700708008    -0.07416992188  1.857836914
-9.055310059    0.01674804688   4.122412109
-9.095385742    -0.05084228516  3.916650391
-7.98762207     -0.01495361328  6.002380371
-7.18371582     -0.03469238281  7.094592285
-7.112536621    -0.02930908203  7.064685059
-6.613085938    -0.03708496094  7.550976563
-5.699121094    -0.07117919922  8.195776367
-5.615380859    -0.01854248047  8.372229004
-4.500439453    -0.125012207    8.991308594
-4.560253906    -0.1321899414   8.960803223
-3.667224121    -0.1668823242   9.371130371
-2.974572754    -0.1357788086   9.594238281
-9.817346191    -0.09450683594  0.8374023438
-9.069665527    0.1220214844    -3.480603027
-9.76829834     -0.2835205078   -0.7279418945
-9.661828613    -0.1932006836   -1.42298584
-9.526647949    -0.06759033203  -2.177844238
```

```
-9.528442383    -0.07117919922  -2.15690918
-9.265856934    -0.07416992188  -2.963208008
-8.679077148    -0.1375732422   -4.43404541
-8.693432617    -0.05443115234  -4.220507813
-8.189794922    -0.02213134766  -5.03996582
-8.212524414    -0.05502929688  -5.221203613
-7.646679688    0.08852539063   -5.903088379
-7.75135498     -0.1674804688   -5.724841309
-6.564038086    -0.3588867188   -7.053918457
-5.823535156    -0.4175048828   -7.601220703
-5.881555176    -0.2709594727   -7.83449707
-5.452087402    -0.1357788086   -8.008557129
-5.252307129    -0.1890136719   -8.220898438
-3.938781738    0.1866210938    -8.921923828
-4.128393555    0.2583984375    -8.622851563
-3.997998047    0.3505126953    -8.746069336
-3.724047852    0.4749267578    -8.834594727
-3.234765625    0.5018432617    -8.965588379
-3.161193848    0.4067382813    -9.060693359
-2.843579102    -0.005383300781 -9.250305176
-1.748376465    -0.4055419922   -9.536816406
-1.801611328    -0.3714477539   -9.368139648
-1.163989258    0.2165283203    -9.246716309
-1.284216309    -0.04545898438  -9.354382324
-0.6340332031   -0.4587768555   -9.43034668
-0.4856933594   -0.3582885742   -9.444104004
0.3582885742    -0.3768310547   -9.634313965
0.2308837891    -0.6130981445   -9.427954102
0.6908569336    -0.5813964844   -9.24552002
1.345227051     -0.7710083008   -9.030786133
1.196887207     -0.6723144531   -9.304736328
2.911767578     -0.5317504883   -9.696520996
2.340539551     -0.5819946289   -9.217407227
3.919641113     -0.8589355469   -8.771191406
4.492663574     -0.8421875      -8.52355957
4.017138672     -0.7859619141   -8.716760254
4.956225586     -0.9241333008   -8.369238281
6.434240723     -1.06529541     -7.232763672
7.186706543     -0.8499633789   -5.900097656
7.742382813     -0.6854736328   -5.947351074
8.509802246     -0.5604614258   -4.259985352
8.789135742     -0.3331665039   -4.249816895
9.242529297     -0.2428466797   -3.353198242
9.40402832      -0.276940918    -2.708398438
9.472814941     -0.1716674805   -2.847167969
```

```
9.076843262     -0.2583984375   -3.363366699
9.579882813     -0.4438232422   -1.735217285
9.789831543     -0.1268066406   -0.3618774414
9.805981445     -0.1327880859   -0.2302856445
9.816748047     -0.1274047852   -0.2368652344
9.792822266     -0.1220214844   -0.09331054688
9.834094238     -0.1118530273   -0.2255004883
9.797607422     -0.133984375    -0.182434082
9.685754395     -0.1208251953   1.520483398
9.585864258     -0.09749755859  2.266967773
9.373522949     -0.09510498047  3.042163086
9.103759766     -0.07058105469  3.829321289
8.878857422     -0.1028808594   4.432849121
8.587561035     -0.07536621094  4.956225586
8.549279785     -0.06639404297  5.029199219
8.308227539     -0.07775878906  5.379711914
7.953527832     -0.05024414063  5.919836426
7.798010254     -0.06220703125  6.136962891
7.66940918      -0.05861816406  6.319396973
7.655651855     -0.03648681641  6.336743164
7.491162109     -0.06041259766  6.538317871
6.96060791      -0.04486083984  7.082629395
6.403735352     -0.04366455078  7.653857422
5.957519531     -0.0478515625   8.048034668
5.426367188     -0.200378418    8.372827148
5.420385742     -0.144152832    8.388977051
5.060302734     -0.006579589844 8.654553223
5.039367676     -0.01495361328  8.631225586
4.112243652     -0.02213134766  9.088208008
4.134375        -0.0005981445313        9.08581543
3.421984863     0.04605712891   9.427954102
1.995410156     0.07416992188   9.93996582
2.043859863     0.09989013672   9.872375488
1.865612793     -0.09929199219  9.884338379
1.27644043      -0.1321899414   9.941760254
0.6442016602    -0.1166381836   10.01114502
0.153125        -0.07177734375  10.08052979
-0.1052734375   -0.07775878906  10.03566895
-0.03768310547  -0.05263671875  10.05959473
-0.019140625    -0.07237548828  10.05181885
-0.019140625    -0.07775878906  10.0847168
-0.01136474609  -0.09869384766  10.07813721
-0.03229980469  -0.1764526367   10.0739502
-0.04127197266  -0.1937988281   10.0739502
-0.02811279297  -0.1967895508   10.08651123
```

```
-0.007775878906 -0.1955932617   10.04404297
-0.03648681641  -0.1854248047   10.07155762
-0.02392578125  -0.2051635742   10.06378174
0.007177734375  -0.162097168    10.06976318
0.03768310547   -0.1471435547   10.05959473
0.03110351563   -0.1447509766   10.05361328
0.06220703125   -0.1704711914   10.05600586
0.05921630859   -0.1375732422   10.09488525
0.1100585938    -0.1656860352   10.05540771
0.08493652344   -0.144152832    10.05361328
0.07596435547   -0.1357788086   10.0793335
0.08433837891   -0.1291992188   10.06138916
0.1208251953    -0.1555175781   10.06138916
0.1393676758    -0.1519287109   10.06378174
-0.1118530273   -0.2512207031   10.10864258
-0.163293457    -0.2877075195   10.06557617
-0.1363769531   -0.2260986328   10.06617432
-0.133984375    -0.2326782227   10.05540771
-0.1782470703   -0.2171264648   10.06796875
-0.1638916016   -0.1878173828   10.06557617
-0.1333862305   -0.1740600586   10.06737061
-0.182434082    -0.1848266602   10.04882813
-0.1878173828   -0.1788452148   10.08352051
-0.2057617188   -0.1890136719   10.06079102
-0.1668823242   -0.1997802734   10.06916504
-0.1686767578   -0.2165283203   10.05720215
-0.1866210938   -0.2518188477   10.07155762
-0.2123413086   -0.3026611328   10.06378174
-0.2177246094   -0.3385498047   10.0739502
-0.2302856445   -0.353503418    10.056604
-0.2404541016   -0.3511108398   10.06916504
-0.2554077148   -0.3600830078   10.05241699
-0.2631835938   -0.3325683594   10.04523926
-0.1710693359   -0.2123413086   10.07215576
-0.1961914063   -0.2392578125   10.06916504
-0.2296875      -0.2542114258   10.06079102
-0.2249023438   -0.2320800781   10.06737061
-0.2673706055   -0.2051635742   10.05600586
-0.2266967773   -0.2147338867   10.07694092
-0.2895019531   -0.2362670898   10.0512207
-0.3116333008   -0.2278930664   10.06737061
-0.3116333008   -0.2344726563   10.05839844
-0.3307739258   -0.2159301758   10.07095947
-0.4306640625   -0.2266967773   10.06976318
-0.4408325195   -0.1955932617   10.07574463
```

```
-0.4264770508   -0.2147338867   10.06079102
-0.4480102539   -0.2033691406   10.06019287
-0.4175048828   -0.2117431641   10.08052979
-0.4480102539   -0.1118530273   10.03626709
-0.1609008789   -0.04127197266  10.07813721
-0.3008666992   -0.1345825195   10.05241699
-0.2350708008   -0.08433837891  10.05002441
-0.2177246094   -0.05981445313  10.06138916
-0.2290893555   -0.06759033203  10.07873535
-0.219519043    -0.06340332031  10.06617432
-0.1519287109   0.00478515625   10.07036133
-0.1764526367   -0.006579589844 10.0619873
-0.2069580078   -0.04306640625  10.07454834
-0.07835693359  0.06938476563   10.07454834
-0.08493652344  0.06340332031   10.03686523
-0.153125       0.09331054688   10.02789307
-0.002990722656 0.09929199219   10.0512207
-0.09031982422  0.03708496094   10.06737061
-0.08134765625  0.02572021484   10.0619873
0.00478515625   0.06459960938   10.05480957
0.0478515625    0.2745483398    10.05959473
-0.09390869141  0.07716064453   10.04224854
-0.08433837891  0.05981445313   10.056604
-0.07596435547  0.04904785156   10.07155762
-0.008972167969 0.04964599609   10.06138916
-0.01136474609  0.0861328125    10.06378174
-0.008374023438 0.06340332031   10.07215576
0.001196289063  0.06818847656   10.05600586
-0.005383300781 0.05443115234   10.05421143
0.04545898438   0.02033691406   10.01174316
0.1866210938    0.02572021484   10.03447266
0.2081542969    0.007177734375  10.04882813
0.1926025391    0.02392578125   10.06378174
0.1519287109    0.005383300781  10.06258545
0.142956543     0.01435546875   10.04404297
0.2123413086    0.004187011719  10.0404541
0.2009765625    -0.01076660156  10.05002441
0.2081542969    0.0005981445313 10.03566895
0.2159301758    -0.04605712891  10.04344482
-2.768811035    -1.853051758    9.536816406
-2.921337891    -2.599536133    8.890222168
-2.956628418    -2.786157227    9.066674805
-3.517687988    -3.922033691    8.591149902
-4.233666992    -5.073461914    7.782458496
-4.664929199    -5.539416504    7.138256836
```

```
-5.313317871    -6.301452637    5.507116699
-5.348608398    -6.439025879    5.645288086
-5.114135742    -6.257189941    5.986828613
-5.408422852    -7.037768555    4.416699219
-5.412609863    -7.567724609    3.68815918
-5.559155273    -7.862011719    2.311828613
-5.310327148    -8.221496582    -0.6071166992
-5.444311523    -8.359667969    -0.08374023438
-5.536425781    -8.254992676    -0.0669921875
-5.615380859    -8.226879883    0.257800293
-5.364758301    -8.207141113    -1.165783691
-5.78046875     -7.026403809    -3.679785156
-5.965893555    -7.430749512    -3.801806641
-5.42277832     -7.508508301    -3.405236816
-5.476013184    -7.418786621    -3.82154541
-4.952636719    -7.137060547    -5.012451172
-4.563244629    -6.359472656    -5.589660645
-4.085925293    -6.07355957     -6.517382813
-3.407629395    -5.873181152    -7.146032715
-3.078051758    -5.550183105    -7.473217773
-2.963208008    -5.30255127     -8.076147461
-2.959020996    -5.318103027    -7.786047363
-2.802905273    -5.047741699    -7.570715332
-2.488879395    -4.675097656    -8.236450195
-2.638415527    -4.941870117    -8.069567871
-1.879370117    -4.117028809    -8.504418945
-1.640710449    -3.634924316    -8.943457031
-0.9354980469   -2.755053711    -9.27722168
-0.04007568359  -0.5909667969   -9.191088867
-0.4815063477   -0.9642089844   -9.990808105
0.2954833984    0.5431152344    -9.148620605
0.4288696289    0.9654052734    -9.533825684
2.734716797     5.061499023     -7.70111084
3.77130127      6.52755127      -5.910864258
4.65715332      6.859521484     -4.520776367
4.642797852     6.819445801     -4.854541016
5.060900879     7.437927246     -2.637817383
4.691845703     8.034875488     -1.047949219
4.731921387     8.329760742     1.619775391
4.692443848     8.235852051     1.513903809
4.789941406     8.469726563     1.303356934
4.428662109     7.888928223     3.436938477
4.283312988     7.877563477     3.040368652
4.254602051     8.352490234     1.915856934
4.275537109     8.480493164     1.673608398
```

```
4.331164551     8.476904297     2.612097168
4.347314453     8.44041748      2.490075684
4.050634766     8.166467285     2.594152832
4.081738281     8.103662109     3.59185791
3.996801758     7.89251709      3.965100098
4.05123291      7.90567627      4.258190918
3.843676758     7.67479248      4.03927002
3.947155762     7.670007324     4.340136719
3.739001465     7.29855957      4.824035645
3.837695313     7.388879395     5.163183594
3.754553223     6.760229492     6.223693848
3.394470215     6.430651855     6.72434082
3.191699219     5.745178223     7.237548828
3.380114746     5.632727051     7.643688965
3.401049805     5.522070313     7.65625
1.221411133     3.31730957      9.403430176
1.366162109     3.163586426     9.562536621
1.187316895     1.570129395     9.699511719
1.242346191     1.653869629     10.02729492
0.9642089844    -0.2117431641   9.733605957
-2.668920898    2.606115723     9.210827637
-3.92442627     2.897412109     8.041455078
-4.090112305    2.656958008     8.297460938
-4.032092285    4.367053223     7.896704102
-4.148132324    4.187609863     7.82791748
-4.577600098    5.217016602     7.104162598
-4.565039063    4.862316895     7.076049805
-4.802502441    5.666223145     6.294274902
-4.8234375      5.706298828     6.281115723
-5.500537109    6.084924316     5.318103027
-5.411413574    6.194384766     5.434143066
-5.194885254    7.231567383     4.108654785
-5.444909668    7.45168457      2.71796875
-6.116625977    7.220202637     0.9348999023
-6.446801758    7.169958496     -0.2117431641
-6.413305664    7.423571777     -0.3427368164
-7.428955078    6.071765137     -1.076660156
-7.193884277    6.331359863     -1.32668457
-7.15201416     5.681774902     -3.052331543
-6.741088867    5.337243652     -3.629541016
-5.571118164    3.959716797     -6.435437012
-4.989123535    1.674206543     -7.826123047
-4.741491699    1.655065918     -8.40871582
-4.454382324    1.35958252      -8.39675293
-4.091906738    0.02572021484   -8.575598145
```

```
-4.154711914    -0.02332763672  -8.563635254
-3.35020752     -3.305944824    -9.003869629
-2.932702637    -3.811975098    -8.304638672
-1.495361328    -5.169165039    -7.928405762
-1.427172852    -5.27623291     -8.14732666
-1.460668945    -5.309729004    -8.286694336
-0.3313720703   -6.460559082    -7.774084473
-0.3301757813   -6.281713867    -7.519274902
0.1525268555    -7.640100098    -6.056811523
0.1920043945    -7.63112793     -6.269152832
1.49296875      -8.138952637    -5.153015137
1.455285645     -8.037268066    -5.125500488
1.577307129     -9.009851074    -3.896313477
1.030004883     -10.0081543     -1.197485352
1.235168457     -9.740783691    -0.05502929688
1.227392578     -9.826916504    0.02572021484
5.863012695     -7.808178711    -2.262182617
5.806787109     -7.853039551    -2.11204834
4.042260742     -7.91763916     -3.990820313
3.17734375      -7.452282715    -5.809179688
3.154016113     -7.204650879    -5.705102539
2.567834473     -6.554467773    -6.706994629
4.027307129     -8.64498291     -3.164782715
6.037670898     -7.591052246    -1.792041016
5.930004883     -7.985229492    0.05443115234
6.046044922     -7.92421875     0.1190307617
5.619567871     -8.223291016    1.470239258
5.56932373      -7.613781738    3.374731445
5.044750977     -7.239941406    4.874279785
5.063891602     -7.242333984    4.849755859
4.448999023     -6.653161621    5.960510254
4.599731445     -6.611889648    6.024511719
6.415698242     -5.952734375    5.083630371
5.460461426     -6.044848633    5.613586426
6.706994629     -5.348010254    5.359375
5.986230469     -5.776281738    5.578295898
5.828320313     -5.86361084     5.526257324
6.704602051     -4.827026367    5.897106934
6.480297852     -4.827026367    5.860620117
6.059204102     -4.727734375    6.901989746
6.156103516     -4.641003418    6.916345215
5.267858887     -4.036877441    7.348803711
5.999389648     -4.19119873     7.474414063
3.909472656     -3.300561523    8.577392578
4.00637207      -3.489575195    8.64498291
```

```
3.814367676     -3.353796387    8.560046387
6.943261719     -2.321398926    6.915148926
4.148132324     -0.9247314453   9.095983887
3.992016602     -1.064099121    9.001477051
3.000891113     -0.410925293    9.761120605
2.636621094     -0.5915649414   9.549975586
2.524169922     -0.3780273438   9.549975586
4.62545166      -0.6202758789   8.869287109
4.709191895     -0.8206542969   8.982336426
1.662841797     0.316418457     9.983630371
1.50612793      0.1178344727    10.01772461
0.5299560547    0.07716064453   10.01174316
-0.9043945313   -0.9989013672   9.846057129];

%% ---------------  [Calibrated Norm Data]  --------------- %%

C1 = [-0.002138592356   -0.007425667903 0.9999701425
-0.003326699221 -0.0114058259   1.000683007
-0.01669290145  -0.3667685891   0.9360499932
0.005108859517  -0.8734367614   0.5019751502
0.00570291295   -0.8575755348   0.5048860121
0.0005940534322 -0.9710991457   0.2386312637
0.008257342708  -0.9933167441   0.02150473425
0.006950425157  -1.002524572    0.03273234412
-0.008970206827 -0.9412776634   -0.2950069345
0.007544478589  -0.8976147361   -0.4035999019
0.00368313128   -0.810704719    -0.5467073737
0.01782160297   -0.8153977411   -0.5412420821
-0.001069296178 -0.5820535529   -0.7719130299
0.01936614189   -0.358630057    -0.8911395537
0.02447500141   -0.3623725937   -0.8904860949
0.0136038236    -0.03736596089  -0.9545844603
-0.02168295028  -0.3592835158   -0.8929811193
-0.07057354775  -0.5563310393   -0.7874772298
-0.07829624237  -0.6939138142   -0.6790624784
-0.09855346441  -0.8332787494   -0.5003118006
-0.09748416823  -0.8374965288   -0.5001335846
-0.09938513921  -0.8743278416   -0.4396589452
-0.03748477157  0.1534440015    -0.936881668
-0.02768288994  0.1628894511    -0.9506637076
-0.02566310827  0.398431637     -0.8400509585
-0.0346333151   0.5260343143    -0.7814772901
-0.01116820453  0.6444291633    -0.6929633287
-0.006178155695 0.6463895396    -0.6936167875
-0.05114800052  0.600409804     -0.7059136935
```

```
0.007722694619  0.7895564168    -0.5088661701
0.0092078282    0.7803485886    -0.5425489997
0.0438411433    0.8708229263    -0.3539370349
0.04520746619   0.926842165     -0.1817209449
0.04657378909   0.9329609153    -0.1783348404
0.04972227228   0.9279708665    -0.176196248
0.01526717321   0.955059703     -0.03611844868
0.009089017513  0.8822881576    0.4020553629
0.01396025566   0.8783079996    0.4055602782
0.01093058315   0.7455964628    0.6327263107
-0.001900970983 0.4911039724    0.8455756555
-0.004811832801 0.4796981465    0.862624989
-0.009089017513 0.2448094194    0.9557131618
0.0005940534322 0.4510053658    0.8777733515
-0.0070098305   0.2234829012    0.977217896
0.007128641187  0.001782160297  0.9973563074
0.005940534322  0.003920752653  0.9978315501
0.006950425157  0.00350491525   0.9992572784
-0.2209284715   -0.02186116631  0.9574953221
-0.2210472821   -0.02667299911  0.9836336731
-0.2257997096   -0.03380164029  0.9902870715
-0.5269847997   -0.009386044229 0.8553181317
-0.5323312806   -0.009683070946 0.8423083616
-0.0828704538   -0.02702943117  0.9996731158
-0.2219977676   -0.02572251362  0.9755545464
-0.4570053054   -0.01045534041  0.8874564224
-0.6396767358   -0.002079187013 0.7599725559
-0.6381321969   -0.003445509907 0.7631210391
-0.6361718206   -0.001247512208 0.7616953108
-0.8423677669   -0.005762318293 0.5176581609
-0.965099206    -0.007247451873 0.1744140877
-0.9680100678   -0.00736626256  0.1714438205
-0.9634358564   -0.00736626256  0.1845129961
-0.8993374911   0.00166334961   0.4094216255
-0.9033176491   -0.005049454174 0.3889861874
-0.7932989534   -0.001485133581 0.5961326193
-0.7134581721   -0.003445509907 0.704606776
-0.7063889363   -0.002910861818 0.7016365088
-0.6567854747   -0.00368313128  0.7499330529
-0.5660141102   -0.007069235844 0.8139720129
-0.5576973622   -0.00184156564  0.8314965891
-0.4469658024   -0.01241571673  0.8929811193
-0.4529063367   -0.01312858085  0.8899514468
-0.3642141593   -0.01657409076  0.9307035123
-0.2954227719   -0.01348501291  0.9528617053
```

```
-0.9750198983   -0.009386044229 0.08316748051
-0.9007632193   0.01211869002   -0.3456796922
-0.9701486602   -0.02815813269  -0.0722963027
-0.9595745091   -0.01918792586  -0.1413253115
-0.9461489015   -0.006712803784 -0.2162948547
-0.9463271176   -0.007069235844 -0.2142156677
-0.9202481719   -0.00736626256  -0.2942940703
-0.8619715302   -0.01366322894  -0.4403718093
-0.8633972584   -0.005405886233 -0.4191641018
-0.8133779594   -0.002197997699 -0.500549422
-0.8156353625   -0.005465291577 -0.518549241
-0.7594379078   0.008791990797  -0.5862713323
-0.7698338428   -0.0166334961   -0.56856854
-0.6519142365   -0.03564320593  -0.7005672126
-0.5783704216   -0.04146492957  -0.7549231017
-0.5841327399   -0.02691062048  -0.7780911856
-0.5414797035   -0.01348501291  -0.7953781404
-0.5216383189   -0.01877208846  -0.8164670373
-0.3911841851   0.01853446709   -0.8860900995
-0.4100156789   0.02566310827   -0.8563874279
-0.3970653141   0.03481153113   -0.8686249286
-0.3698576669   0.04716784252   -0.8774169194
-0.3212640962   0.04984108297   -0.8904266896
-0.3139572389   0.04039563339   -0.8998721392
-0.2824130017   -0.000534648089 -0.918703633
-0.1736418182   -0.04027682271  -0.9471587924
-0.1789288938   -0.03689071814  -0.9304064856
-0.1156027979   0.02150473425   -0.9183472009
-0.1275432719   -0.004514806085 -0.9290401627
-0.06296966382  -0.04556389825  -0.9365846413
-0.0482371387   -0.03558380059  -0.9379509642
0.03558380059   -0.03742536623  -0.9568418633
0.02293046248   -0.0608904768   -0.9363470199
0.06861317142   -0.05774199361  -0.9182283902
0.1336026169    -0.07657348742  -0.896901872
0.1188700918    -0.06677160578  -0.9241095192
0.2891852108    -0.05281135013  -0.963020019
0.232453108     -0.05780139896  -0.9154363391
0.3892832141    -0.08530607287  -0.871119953
0.446193533     -0.08364272326  -0.8465261409
0.3989662851    -0.078058621    -0.8657140668
0.492232674     -0.09178125528  -0.8311995624
0.6390232771    -0.1058009163   -0.7183294103
0.7137551988    -0.08441499272  -0.5859743056
0.7689427627    -0.06807852333  -0.5906673277
```

```
0.8451598181    -0.0556628066   -0.4230848544
0.8729021133    -0.03308877618  -0.4220749636
0.9179313635    -0.02411856935  -0.3330263541
0.9339708062    -0.02750467391  -0.2689873941
0.9408024206    -0.01704933351  -0.2827694337
0.9014760834    -0.02566310827  -0.3340362449
0.9514359771    -0.04407876467  -0.1723349007
0.9722872526    -0.01259393276  -0.03594023265
0.9738911968    -0.0131879862   -0.02287105714
0.974960493     -0.01265333811  -0.02352451592
0.9725842793    -0.01211869002  -0.009267233543
0.9766832479    -0.01110879918  -0.0223958144
0.973059522     -0.01330679688  -0.01811862968
0.9619507228    -0.01199987933  0.1510083825
0.9520300305    -0.009683070946 0.2251462508
0.9309411337    -0.009445449573 0.3021355756
0.9041493239    -0.0070098305   0.3803130073
0.8818129148    -0.01021771903  0.4402529986
0.8528825127    -0.007485073246 0.492232674
0.8490805707    -0.006593993098 0.4994801258
0.8251402174    -0.007722694619 0.534291657
0.7899128489    -0.004990048831 0.5879346819
0.7744674596    -0.006178155695 0.6094988215
0.7616953108    -0.005821723636 0.6276174512
0.7603289879    -0.003623725937 0.6293402061
0.7439925185    -0.005999939666 0.6493598068
0.6912999791    -0.004455400742 0.7034186691
0.6359936046    -0.004336590055 0.7601507719
0.5916772185    -0.004752427458 0.7992988931
0.5389252737    -0.01990078998  0.8315559945
0.5383312203    -0.01431668772  0.8331599387
0.5025692037    -0.0006534587755        0.8595359111
0.5004900167    -0.001485133581 0.8572191027
0.4084117347    -0.002197997699 0.9026047849
0.4106097324    -0.00005940534322       0.9023671636
0.3398579686    0.004574211428  0.9363470199
0.198176225     0.00736626256   0.9871979937
0.2029880578    0.009920692318  0.9804851899
0.1852852655    -0.009861286975 0.9816732968
0.1267710024    -0.01312858085  0.9873762097
0.06397955465   -0.01158404193  0.9942672295
0.01520776787   -0.007128641187 1.001158249
-0.01045534041  -0.007722694619 0.9967028486
-0.003742536623 -0.005227670204 0.9990790623
-0.001900970983 -0.00718804653  0.9983067929
```

```
-0.001900970983 -0.007722694619 1.001574087
-0.001128701521 -0.009801881632 1.000920628
-0.003207888534 -0.01752457625  1.000504791
-0.004098968682 -0.0192473312   1.000504791
-0.002792051132 -0.01954435792  1.001752303
-0.0007722694619        -0.01942554723  0.9975345234
-0.003623725937 -0.0184156564   1.000267169
-0.002376213729 -0.02037603273  0.9994948997
0.0007128641187 -0.01609884801  1.000088953
0.003742536623  -0.01461371443  0.9990790623
0.003089077848  -0.01437609306  0.9984850089
0.006178155695  -0.01693052282  0.9987226303
0.005881128979  -0.01366322894  1.002583978
0.01093058315   -0.01645528007  0.9986632249
0.008435558738  -0.01431668772  0.9984850089
0.007544478589  -0.01348501291  1.001039439
0.008376153395  -0.01283155414  0.9992572784
0.01199987933   -0.01544538924  0.9992572784
0.01384144497   -0.01508895718  0.9994948997
-0.01110879918  -0.02495024415  1.0039503
-0.0162176587   -0.02857397009  0.9996731158
-0.01354441826  -0.02245521974  0.9997325211
-0.01330679688  -0.02310867851  0.9986632249
-0.01770279228  -0.02156413959  0.9999107372
-0.01627706404  -0.01865327777  0.9996731158
-0.01324739154  -0.01728695488  0.9998513318
-0.01811862968  -0.01835625106  0.9980097662
-0.01865327777  -0.01776219762  1.001455276
-0.02043543807  -0.01877208846  0.999197873
-0.01657409076  -0.01984138464  1.000029548
-0.01675230679  -0.02150473425  0.998841441
-0.01853446709  -0.0250096495   1.000267169
-0.02108889684  -0.03005910367  0.9994948997
-0.02162354493  -0.03362342426  1.000504791
-0.02287105714  -0.03510855785  0.9987820356
-0.02388094798  -0.03487093647  1.000029548
-0.02536608156  -0.03576201662  0.9983661982
-0.02613835102  -0.03302937083  0.9976533341
-0.01698992816  -0.02108889684  1.000326575
-0.01948495258  -0.02376213729  1.000029548
-0.0228116518   -0.02524727087  0.999197873
-0.02233640905  -0.02304927317  0.9998513318
-0.02655418842  -0.02037603273  0.9987226303
-0.02251462508  -0.02132651822  1.000801817
-0.02875218612  -0.02346511057  0.9982473875
```

```
-0.03095018382  -0.02263343577  0.9998513318
-0.03095018382  -0.02328689454  0.9989602517
-0.0328511548   -0.0214453289   1.000207764
-0.04277184712  -0.02251462508  1.000088953
-0.04378173796  -0.01942554723  1.000683007
-0.04235600972  -0.02132651822  0.999197873
-0.04449460207  -0.0201978167   0.9991384677
-0.04146492957  -0.0210294915   1.001158249
-0.04449460207  -0.01110879918  0.996762254
-0.01598003733  -0.004098968682 1.000920628
-0.02988088764  -0.01336620223  0.9983661982
-0.02334629989  -0.008376153395 0.9981285769
-0.02162354493  -0.005940534322 0.9992572784
-0.02275224645  -0.006712803784 1.000980033
-0.02180176096  -0.006296966382 0.9997325211
-0.01508895718  0.0004752427458 1.000148359
-0.01752457625  -0.0006534587755        0.9993166837
-0.02055424876  -0.004277184712 1.000564196
-0.007782099962 0.006891019814  1.000564196
-0.008435558738 0.006296966382  0.9968216593
-0.01520776787  0.009267233543  0.9959305792
-0.0002970267161        0.009861286975  0.9982473875
-0.008970206827 0.00368313128   0.9998513318
-0.008079126678 0.002554429759  0.9993166837
0.0004752427458 0.006415777068  0.9986038196
0.004752427458  0.02726705254   0.9990790623
-0.009326638886 0.007663289276  0.9973563074
-0.008376153395 0.005940534322  0.9987820356
-0.007544478589 0.004871238144  1.000267169
-0.0008910801484        0.004930643488  0.9992572784
-0.001128701521 0.008554369424  0.9994948997
-0.0008316748051        0.006296966382  1.000326575
0.0001188106864 0.006772209128  0.9987226303
-0.000534648089 0.005405886233  0.9985444143
0.004514806085  0.00201978167   0.9943266349
0.01853446709   0.002554429759  0.9965840379
0.02067305944   0.0007128641187 0.9980097662
0.01912852052   0.002376213729  0.9994948997
0.01508895718   0.000534648089  0.9993760891
0.01419787703   0.001425728237  0.9975345234
0.02108889684   0.0004158374026 0.9971780914
0.01996019532   -0.001069296178 0.9981285769
0.02067305944   0.00005940534322        0.9967028486
0.0214453289    -0.004574211428 0.9974751181
-0.2749873338   -0.1840377533   0.9471587924
```

```
-0.2901356963   -0.2581756217   0.8829416163
-0.2936406116   -0.2767100887   0.9004661926
-0.3493628235   -0.3895208355   0.8532389447
-0.4204710193   -0.5038761212   0.7729229207
-0.4633022718   -0.5501528836   0.708943366
-0.5276976639   -0.6258352909   0.5469449951
-0.5312025791   -0.6394985198   0.5606676294
-0.5079156846   -0.6214392955   0.5945880803
-0.5371431134   -0.6989632684   0.4386490544
-0.5375589508   -0.7515964025   0.3662933463
-0.5521132599   -0.7808238313   0.2296016516
-0.5274006371   -0.8165264426   -0.06029642337
-0.540707434    -0.8302490769   -0.008316748051
-0.5498558569   -0.8198531418   -0.006653398441
-0.5576973622   -0.8170610907   0.02560370293
-0.5328065234   -0.8151007144   -0.1157810139
-0.5740932369   -0.6978345669   -0.3654616715
-0.5925088933   -0.7379925789   -0.3775803615
-0.5385688417   -0.7457152735   -0.338194619
-0.5438559172   -0.736804472    -0.3795407379
-0.4918762419   -0.7088245554   -0.4978167762
-0.4532033635   -0.6315976092   -0.5551429324
-0.4057978996   -0.6032018551   -0.6472806198
-0.3384322403   -0.5833010651   -0.7097156355
-0.3056998962   -0.5512221798   -0.7422103582
-0.2942940703   -0.5266283677   -0.8020909442
-0.2938782329   -0.5281729066   -0.7732793528
-0.2783734383   -0.5013216915   -0.7518934292
-0.2471856332   -0.4643121626   -0.8180115762
-0.262036969    -0.4908069457   -0.8014374854
-0.1866515884   -0.4088869774   -0.84462517
-0.1629488565   -0.3610062708   -0.8882286919
-0.0929099568   -0.2736210109   -0.9213768734
-0.003980157996 -0.05869247911  -0.912822504
-0.0478213013   -0.09576141328  -0.9922474479
0.02934623955   0.05394005165   -0.9086047246
0.04259363109   0.09588022396   -0.9468617657
0.2716012292    0.5026880144    -0.764843794
0.374550689     0.6482905106    -0.5870436017
0.4625300023    0.6812604761    -0.4489855841
0.4611042741    0.6772803181    -0.4821337656
0.502628609     0.738705443     -0.2619775636
0.4659755123    0.7979919755    -0.1040781613
0.4699556702    0.8272788097    0.1608696695
0.4660349176    0.8179521709    0.1503549237
```

```
0.4757179885    0.8411796601    0.1294442429
0.4398371612    0.7834970718    0.3413431022
0.4254016628    0.7823683703    0.3019573596
0.4225502064    0.8295362128    0.1902753143
0.4246293934    0.8422489562    0.1662161503
0.4301540903    0.8418925242    0.2594231339
0.4317580346    0.8382687982    0.2473044438
0.4022929843    0.811061151     0.2576409736
0.4053820622    0.80482359      0.3567290861
0.3969465034    0.7838535038    0.3937980202
0.4023523897    0.7851604214    0.4229066384
0.3817387356    0.7622299589    0.4011642828
0.3920158599    0.7617547162    0.4310451704
0.3713428005    0.724863998     0.4791040931
0.3811446821    0.7338342048    0.5127869227
0.3728873394    0.6713991891    0.6181125962
0.3371253228    0.638666845     0.6678348685
0.3169869114    0.5705883217    0.718804653
0.3356995946    0.5594201171    0.7591408811
0.3377787816    0.5484301286    0.7603883933
0.1213057109    0.3294620335    0.9339114008
0.1356818039    0.3141948603    0.9497132221
0.1179196063    0.155939026     0.9633170457
0.1233848979    0.164255774     0.9958711738
0.09576141328   -0.0210294915   0.9667031503
-0.2650666415   0.2588290804    0.9147828803
-0.3897584569   0.2877594826    0.7986454343
-0.406213737    0.2638785346    0.8240709212
-0.4004514187   0.4337184109    0.7842693412
-0.4119760553   0.4158968079    0.7774377268
-0.4546290917   0.5181334036    0.7055572615
-0.4533815795   0.4829060351    0.7027652103
-0.4769655007   0.5627468164    0.6251224267
-0.4790446878   0.5667269744    0.6238155092
-0.5462915363   0.6043305566    0.5281729066
-0.5374401401   0.6152017344    0.5396975432
-0.5159354059   0.7182105996    0.4080553026
-0.5407668394   0.7400717659    0.2699378796
-0.6074790398   0.7170818981    0.09285055146
-0.6402707893   0.7120918492    -0.0210294915
-0.6369440901   0.7372797148    -0.03403926167
-0.7378143628   0.6030236391    -0.1069296178
-0.714468063    0.628805558     -0.1317610513
-0.7103096889   0.5642913553    -0.3031454665
-0.6694982181   0.5300738776    -0.3604716227
```

```
-0.5533013668   0.3932633721    -0.6391420877
-0.4954999678   0.1662755557    -0.7772595107
-0.4709061557   0.1643745847    -0.835120315
-0.442391591    0.1350283451    -0.8339322082
-0.406391953    0.002554429759  -0.8516944058
-0.412629514    -0.002316808386 -0.8505062989
-0.3327293274   -0.328333332    -0.8942286316
-0.2912643978   -0.3785902524   -0.8247837853
-0.1485133581   -0.5133809761   -0.7874178244
-0.1417411489   -0.5240145326   -0.8091601801
-0.1450678482   -0.5273412318   -0.823001625
-0.03291056015  -0.6416371122   -0.7720912459
-0.03279174946  -0.6238749145   -0.7467845697
0.01514836252   -0.758784449    -0.6015385055
0.01906911517   -0.7578933689   -0.6226274023
0.1482757367    -0.8083285053   -0.5117770319
0.1445332001    -0.7982295969   -0.5090443861
0.1566518901    -0.894822685    -0.3869664058
0.102296001     -0.9939702028   -0.1189294971
0.1226720338    -0.9674160144   -0.005465291577
0.1218997643    -0.9759703838   0.002554429759
0.5822911743    -0.7754773504   -0.2246710081
0.576707072     -0.7799327512   -0.2097602669
0.4014613095    -0.7863485283   -0.39635245
0.3155611832    -0.7401311712   -0.5769446934
0.3132443748    -0.7155373591   -0.5666081637
0.2550271385    -0.6509637511   -0.6661121136
0.3999761759    -0.8585854256   -0.314313671
0.5996375345    -0.7539132109   -0.1779784083
0.5889445727    -0.793061332    0.005405886233
0.6004692093    -0.787001987    0.0118216633
0.5581131996    -0.8167046586   0.1460183336
0.5531231508    -0.7561706139   0.3351649465
0.5010246648    -0.7190422744   0.4840941419
0.5029256357    -0.7192798958   0.4816585229
0.4418569429    -0.6607656327   0.5919742452
0.4568270894    -0.656666664    0.598330617
0.6371817114    -0.5912019758   0.5048860121
0.5423113783    -0.6003503986   0.5575191462
0.6661121136    -0.5311431738   0.5322718753
0.594528675     -0.5736773995   0.5540142309
0.5788456644    -0.5823505796   0.548845966
0.6658744922    -0.4794011198   0.5856772788
0.6435974885    -0.4794011198   0.5820535529
0.6017761269    -0.4695398328   0.6854782555
```

```
0.6113997925     -0.4609260581    0.6869039837
0.5231828578     -0.4009266614    0.7298540469
0.5958355925     -0.41625324      0.7423291689
0.3882733233     -0.3277986839    0.8518726218
0.3978969889     -0.3465707724    0.8585854256
0.3788278737     -0.3330857595    0.8501498669
0.6895772241     -0.2305521371    0.686785173
0.4119760553     -0.09184066062   0.9033770544
0.3964712607     -0.1056821056    0.8939910102
0.298036607      -0.0408114708    0.9694357961
0.2618587529     -0.05875188445   0.9484657099
0.2506905484     -0.03754417692   0.9484657099
0.4593815192     -0.06160334092   0.8808624293
0.4676982672     -0.0815041309    0.8920900392
0.1651468542     0.03142542657    0.9915345838
0.1495826542     0.01170285262    0.9949206883
0.0526331341     0.007663289276   0.9943266349
-0.08982087896   -0.09920692318   0.9778713548];

%% ----------------  [Magneto Data]  ---------------- %%

A = [0.101925 , -0.000025 , -0.000179;
    -0.000025 , 0.102289 , -0.000287;
    -0.000179 , -0.000287 , 0.101810];

B = [-0.002326 , -0.226165 , 0.235843];

D = (A * (C - B)')';

MagD = D(:,1).^2 + D(:,2).^2 + D(:,3).^2;

%% --------------------  [Plot]  -------------------- %%

figure;
hold on;
grid on;
p1 = plot3(C1(:,1),C1(:,2),C1(:,3),'.','MarkerSize', 15);
p2 = plot3(D(:,1),D(:,2),D(:,3),'.','MarkerSize', 15);
ellipsoid(C1(:,1),C1(:,2),C1(:,3),[0 0.4470 0.7410],0.1);
ellipsoid(D(:,1),D(:,2),D(:,3),[0.8500 0.3250 0.0980],0.1);
legend('Uncalibrated Accelerometer Data','Calibrated
   Accelerometer Data');
xlabel('X'), ylabel('Y'), zlabel('Z');
title('Magnetometer Data 3D Scatterplot');
view(45,10);
```

```
axis vis3d;

function ellipsoid(x,y,z,color,trans)

% Matlab Add-on for Ellipsoid Fit by Yury %
%<https://www.mathworks.com/matlabcentral/fileexchange/24693-
   ellipsoid-fit>
[ center, radii, evec, v, chi ] = ellipsoid_fit( [ x y z ], ''
   );

size_min = min( [ x y z ] );
size_max = max( [ x y z ] );

steps = 100;
step = ( size_max - size_min ) / steps;

[ x, y, z ] = meshgrid( linspace( size_min(1) - step(1),
   size_max(1) + step(1), steps ), linspace( size_min(2) - step
   (2), size_max(2) + step(2), steps ), linspace( size_min(3) -
    step(3), size_max(3) + step(3), steps ) );

E = v(1) *x.*x +   v(2) * y.*y + v(3) * z.*z + ...
          2*v(4) *x.*y + 2*v(5)*x.*z + 2*v(6) * y.*z + ...
          2*v(7) *x    + 2*v(8)*y    + 2*v(9) * z;
e = patch( isosurface( x, y, z, E, -v(10) ) );
set( e, 'FaceColor', color      , 'EdgeColor', 'none','
   FaceAlpha',trans);
end
```

Appendix K — MATLAB Code: IMU Raw Data and Calibration for Magnetometer Scatterplot

```
%% -------  [Calibration for Magnetometer Scaling]  ------- %%
clc; clear all; close all;
% By: Jason Nguyen

%% --------------  [Raw Magnetometer Data]  -------------- %%

C = [-14.7       17.7     6.3
-15.15   18       7.05
-15.3    3        11.4
-16.8    -26.7    6.6
-16.95   -27.75   6.3
-15.3    -40.05   -1.05
-16.2    -46.35   -8.25
-18      -45.45   -9.45
-16.65   -51.9    -25.2
-15.45   -52.65   -25.8
-16.2    -51.45   -34.65
-16.5    -52.8    -33.75
-14.55   -49.8    -50.7
-16.65   -42.9    -58.65
-16.95   -42.15   -58.2
-16.95   -32.85   -66.3
-16.2    -44.1    -58.65
-13.2    -48.75   -50.85
-12      -49.8    -43.05
-12.15   -53.1    -34.65
-10.95   -51.9    -32.85
-13.05   -51.6    -30.15
-13.65   -24.6    -70.35
-13.8    -23.25   -70.95
-15.3    -11.55   -73.95
-15      -4.95    -73.5
-15.9    2.1      -73.65
-15.15   0.45     -72.9
-14.55   2.25     -73.2
-17.4    11.85    -71.55
-16.05   12       -70.95
-16.5    21.75    -65.1
-18      26.4     -62.1
-16.65   26.55    -58.65
```

```
-17.7   26.55   -61.95
-15.9   30.15   -54.9
-14.7   38.7    -36.9
-13.95  37.5    -35.85
-15     39.6    -24.9
-13.65  35.1    -9.3
-14.55  33.75   -10.2
-15     27.9    -1.35
-14.25  33.75   -7.95
-15.75  28.35   -0.6
-14.55  19.05   6.15
-15.45  18.45   6
-14.55  18.15   5.4
-8.25   18.6    3.9
-7.05   17.85   5.1
-7.65   16.8    5.7
4.05    18.6    -3.9
4.05    18.6    -3.6
-12.45  18      5.7
-7.2    17.25   5.25
1.2     18.75   -0.15
7.5     18.6    -6.6
6.3     18.6    -6.15
6.3     19.05   -6.15
12.45   19.5    -16.65
14.55   17.1    -30.45
15.9    17.7    -30.6
15      17.85   -31.2
12.9    18      -22.2
14.7    16.05   -22.5
11.7    17.25   -14.1
9       16.65   -9.45
9.45    18.45   -10.35
6.45    19.2    -6.75
5.4     18.45   -2.7
4.2     18.45   -3.75
2.25    17.1    -1.8
1.5     18.15   -0.6
-0.45   16.5    0.9
-3.3    17.1    3.6
15.9    17.7    -36.15
9.45    16.8    -50.85
15.45   12.9    -43.2
14.25   16.2    -44.85
13.8    16.65   -46.65
```

```
13.5    16.65   -48.15
10.8    15.15   -50.7
9       15.45   -55.05
8.4     15.3    -54.6
5.55    16.65   -58.5
5.25    15.45   -58.05
3.45    14.55   -59.7
4.2     15.75   -60.75
-2.4    14.1    -64.2
-3.9    14.1    -65.55
-4.2    13.35   -67.8
-6.3    13.95   -65.85
-7.05   13.95   -67.65
-11.55  17.1    -67.95
-13.95  14.85   -67.05
-11.4   16.05   -68.55
-13.65  17.25   -67.5
-15.45  16.05   -68.25
-15.15  15.3    -68.25
-16.5   15.9    -69.15
-21.3   13.95   -70.2
-20.25  14.7    -69.15
-21.3   13.95   -70.2
-21.15  13.8    -70.05
-24.45  14.55   -70.8
-25.65  14.85   -72.3
-28.8   14.7    -70.65
-30.75  12.75   -69.15
-28.8   11.55   -70.65
-33.75  12      -67.65
-33.9   11.4    -68.55
-38.55  13.5    -65.85
-35.7   12.9    -68.25
-42.15  12.6    -63.75
-43.5   12.45   -64.35
-42     11.25   -63.9
-43.2   11.4    -64.8
-50.55  11.25   -60.6
-54.3   11.55   -52.5
-54.9   14.85   -51.3
-56.7   14.1    -44.7
-58.5   14.25   -43.05
-58.65  14.85   -39.15
-59.25  14.25   -37.5
-58.8   14.25   -36.75
```

```
-57      14.25    -39.3
-58.8    14.25    -33.45
-58.05   15.15    -24.6
-57.75   15.45    -24.15
-58.95   14.85    -26.55
-60.15   16.05    -26.55
-57.9    15.3     -25.5
-57.75   15.9     -25.35
-58.5    15.45    -20.1
-56.25   15.15    -16.5
-55.65   16.8     -14.85
-52.65   16.05    -13.35
-53.4    16.5     -11.1
-51.75   17.4     -9.45
-51.9    15.75    -8.55
-49.8    14.7     -8.4
-48.15   16.8     -4.95
-47.7    16.5     -6
-45.45   16.05    -4.2
-46.5    17.1     -3.9
-46.95   17.25    -4.5
-43.2    15.3     -2.25
-41.4    16.05    -0.9
-40.05   15.9     0
-39      17.85    2.1
-39      16.65    2.85
-34.65   16.8     2.25
-35.55   17.1     1.8
-32.55   15.45    3.9
-32.25   17.85    3.6
-28.05   17.1     5.85
-25.2    17.7     6
-23.55   17.4     6.15
-23.4    16.05    4.5
-20.25   17.4     5.4
-18.45   17.7     4.95
-16.2    17.25    5.25
-15      17.4     4.65
-11.25   18.15    6.15
-13.65   18.75    5.25
-16.65   19.5     6
-15.15   20.7     5.25
-19.5    21.3     7.35
-21.15   19.2     6.3
-20.7    19.95    4.95
```

```
-25.5   21.3    6.6
-25.65  20.7    4.95
-24.75  19.8    5.1
-27.3   20.55   6.15
-28.95  19.95   4.95
-29.7   19.95   5.25
-31.65  17.25   5.25
-33.75  18.6    5.85
-34.65  17.25   4.95
-37.2   15.3    4.5
-38.4   14.85   6.3
-37.5   13.5    4.65
-39.9   11.4    4.2
-39.9   12.15   5.25
-42     6.9     6.15
-42.45  7.95    6.3
-43.65  5.25    6.75
-43.05  4.95    7.35
-45     3.45    5.85
-44.4   1.35    6.3
-44.4   1.65    5.7
-44.4   -1.35   7.05
-45     -1.5    6.6
-45.9   -2.85   5.25
-44.55  -3.75   5.4
-45.9   -4.35   5.25
-45.75  -9.3    7.35
-45.9   -14.1   4.95
-43.5   -16.65  5.85
-41.4   -20.4   6.75
-41.55  -19.05  5.85
-39.3   -22.95  6.15
-37.2   -23.85  6
-36.75  -25.2   6.15
-36.9   -25.8   6
-34.5   -28.35  6.15
-33.3   -29.1   6.15
-30     -29.1   5.85
-31.65  -28.95  6.3
-28.65  -29.7   6
-27.75  -30.6   6.6
-27.3   -29.85  6.6
-25.8   -29.55  6.15
-22.65  -30.15  6.3
-22.05  -31.5   7.35
```

```
-18.45   -30.45   6.3
-19.95   -29.25   7.8
-18.3    -29.55   6.9
-14.7    -27.75   7.5
-9.9     -27.3    7.05
-7.5     -26.4    5.85
-4.2     -22.65   7.05
-0.9     -16.8    7.05
-3.9     -19.5    5.25
-2.25    -18.6    5.1
1.2      -15.75   6
0.15     -14.55   6.3
1.05     -10.5    5.85
2.4      -3.6     6.3
0.75     -3       8.4
0.75     -6.6     5.1
-0.9     6        6
0.45     2.4      4.95
0.3      1.8      6.15
-2.25    8.1      5.85
1.5      1.05     6.6
2.25     -2.7     5.85
1.5      -4.65    5.85
1.2      -5.25    6
0        4.2      5.85
-0.75    4.95     7.35
0.15     9        6
-0.45    7.95     5.7
-2.4     11.85    5.7
-6.15    13.35    5.25
-7.8     15.9     7.65
-9.15    16.05    6
-9       16.65    6.15
-8.7     16.5     8.55
-9       15.9     6.6
-10.65   17.25    6
-12.15   18.75    6.75
-13.05   18.15    5.85
-13.5    18.15    6.6
-3       12       6.15
0        8.85     5.4
-1.8     7.65     5.85
0.15     3.3      6.3
3.9      -2.4     5.25
6.9      -3.6     0.45
```

```
10.2    -11.4   -1.65
10.95   -10.95  -2.25
8.7     -8.25   -0.9
10.8    -15.15  -6.15
12      -17.1   -6.6
13.5    -20.55  -9.45
13.5    -27.15  -20.55
13.05   -25.65  -20.1
13.05   -23.25  -17.25
14.25   -25.2   -18.3
14.55   -26.7   -22.8
18      -21.3   -37.8
17.4    -21.15  -38.7
16.2    -23.4   -37.2
16.5    -25.2   -36.3
14.55   -24     -43.05
14.55   -23.25  -47.1
12.45   -22.65  -49.8
9.9     -23.4   -52.8
11.25   -22.95  -53.7
10.5    -21     -56.25
9.75    -20.55  -55.05
10.2    -23.1   -55.35
9.3     -20.55  -57
8.55    -21.3   -57.75
6.45    -19.5   -59.4
5.25    -17.1   -62.1
4.65    -15.75  -64.2
0.75    -10.05  -66.9
2.25    -8.85   -65.85
0.45    -4.35   -67.5
-0.75   4.65    -66.9
-9.3    13.95   -65.85
-14.55  21.75   -62.1
-19.5   26.7    -58.05
-19.95  26.55   -57.9
-21.45  30.45   -52.05
-22.2   37.05   -45.75
-23.4   38.25   -36.6
-22.2   38.7    -37.2
-23.4   38.25   -35.85
-21.75  40.8    -29.4
-22.2   38.7    -28.8
-20.4   40.2    -34.65
-20.4   37.8    -35.1
```

```
-22.65  37.8    -32.85
-22.5   38.1    -32.25
-21.45  39       -30.3
-20.7   38.7    -28.05
-22.2   37.8    -27
-21     38.4    -26.1
-21.9   39.45   -24.45
-20.4   39       -22.95
-22.05  39.6    -21.6
-21.45  39       -18.9
-21.9   37.8    -15.6
-19.65  37.5    -14.85
-19.8   36.15   -10.65
-19.65  34.8    -8.55
-21.15  34.8    -9.75
-7.8    23.7    1.8
-9.3    24       1.35
-8.4    18       4.95
-7.95   18       6.3
-9.15   10.2    9.3
-4.2    27       -2.25
0.3     25.65   -10.2
-0.45   26.55   -9
0       29.55   -15
0       29.1    -14.25
1.2     31.2    -20.4
0       31.05   -19.8
1.2     31.95   -22.95
2.25    31.8    -23.85
2.7     30.45   -29.55
4.5     28.65   -28.2
2.25    32.25   -35.55
2.4     30.45   -41.7
4.5     24.45   -49.2
3.15    19.95   -54.15
3.3     20.55   -52.95
5.4     12.6    -57.9
4.2     12.9    -57.45
3.45    6       -62.7
1.5     1.8     -65.4
-10.2   -4.05   -72.9
-14.4   -16.8   -73.65
-16.05  -16.2   -72.75
-15.9   -19.2   -72.6
-18.3   -24.45  -70.65
```

```
-19.05  -24.9   -69.45
-24.3   -39.15  -63.15
-22.8   -43.05  -61.35
-27.9   -46.05  -54.6
-28.95  -46.05  -53.85
-28.95  -48     -53.85
-31.8   -49.35  -47.7
-30.45  -48.75  -48
-32.25  -52.05  -37.8
-31.2   -52.65  -38.7
-37.05  -48.6   -32.7
-38.7   -50.25  -33.3
-36     -50.85  -26.1
-32.25  -48.15  -13.5
-28.5   -45     -6.9
-29.4   -44.4   -7.8
-54.9   -34.8   -21.15
-55.8   -34.95  -19.5
-53.25  -39.6   -28.95
-49.65  -43.35  -39.15
-49.8   -42.75  -39
-47.25  -41.55  -44.1
-49.05  -42     -24.6
-54.75  -34.2   -19.05
-54     -31.8   -10.65
-53.25  -30.6   -9.9
-49.8   -28.65  -5.85
-48.15  -22.65  2.7
-45.45  -18.45  4.2
-44.25  -19.05  4.35
-42.15  -14.55  8.55
-40.05  -15.45  7.35
-45.9   -12.15  4.5
-44.4   -11.85  7.05
-45.45  -9.9    5.25
-44.25  -10.8   6.6
-43.95  -12.15  7.05
-44.55  -7.35   7.95
-43.8   -8.1    7.65
-39.45  -6.3    8.55
-39.9   -6.3    10.8
-38.25  -3.75   11.4
-36.3   -3.75   10.65
-30.15  2.1     11.55
-30.15  1.35    13.05
```

```
-29.55  2.7     10.95
-40.2   4.05    10.35
-27.9   10.95   9
-27.9   10.95   10.35
-21.15  14.85   9.3
-23.4   16.05   9.3
-22.95  14.1    9
-27.15  12.6    10.35
-27.15  10.95   10.05
-19.2   19.2    8.25
-20.7   18.45   6.75
-18.45  18.75   7.5
-12.75  14.25   7.65];

%% ---------------  [Calibrated Norm Data]  --------------- %%

C1 = [-0.6162181547     0.7419769618    0.2640934949
-0.6350819758   0.7545528425    0.2955331966
-0.6413699161   0.1257588071    0.4778834669
-0.7042493196   -1.119253383    0.2766693756
-0.71053726     -1.163268965    0.2640934949
-0.6413699161   -1.678880075    -0.04401558248
-0.6790975582   -1.942973569    -0.3458367195
-0.7545528425   -1.905245927    -0.3961402423
-0.6979613793   -2.175627362    -1.056373979
-0.6476578565   -2.207067064    -1.081525741
-0.6790975582   -2.156763541    -1.452514222
-0.6916734389   -2.213355005    -1.41478658
-0.6099302143   -2.087596198    -2.12532384
-0.6979613793   -1.798350941    -2.458584678
-0.71053726     -1.766911239    -2.439720857
-0.71053726     -1.377058938    -2.779269636
-0.6790975582   -1.848654464    -2.458584678
-0.5533387512   -2.043580615    -2.13161178
-0.5030352283   -2.087596198    -1.804638882
-0.5093231687   -2.225930885    -1.452514222
-0.4590196458   -2.175627362    -1.377058938
-0.5470508108   -2.163051482    -1.263876011
-0.5722025722   -1.031222218    -2.949044026
-0.5784905126   -0.9746307549   -2.974195787
-0.6413699161   -0.4841714073   -3.099954595
-0.6287940354   -0.2075020317   -3.081090773
-0.6665216775   0.08803116496   -3.087378714
-0.6350819758   0.01886382106   -3.055939012
-0.6099302143   0.09431910531   -3.068514893
```

```
-0.7294010811   0.496747288     -2.999347549
-0.6728096179   0.5030352283    -2.974195787
-0.6916734389   0.9117513513    -2.728966114
-0.7545528425   1.106677502     -2.603207307
-0.6979613793   1.112965443     -2.458584678
-0.7419769618   1.112965443     -2.596919366
-0.6665216775   1.263876011     -2.30138617
-0.6162181547   1.622288611     -1.546833327
-0.5847784529   1.571985088     -1.502817745
-0.6287940354   1.660016253     -1.043798099
-0.5722025722   1.471378043     -0.3898523019
-0.6099302143   1.41478658      -0.4275799441
-0.6287940354   1.169556906     -0.05659146319
-0.5973543336   1.41478658      -0.3332608388
-0.6602337372   1.188420727     -0.02515176142
-0.6099302143   0.798568425     0.2578055545
-0.6476578565   0.7734166635    0.2515176142
-0.6099302143   0.7608407828    0.2263658527
-0.3458367195   0.7797046039    0.1634864492
-0.2955331966   0.7482649021    0.213789972
-0.3206849581   0.7042493196    0.2389417335
0.1697743896    0.7797046039    -0.1634864492
0.1697743896    0.7797046039    -0.1509105685
-0.5218990494   0.7545528425    0.2389417335
-0.301821137    0.7231131407    0.2200779124
0.05030352283   0.7859925442    -0.006287940354
0.3143970177    0.7797046039    -0.2766693756
0.2640934949    0.7797046039    -0.2578055545
0.2640934949    0.798568425     -0.2578055545
0.5218990494    0.817432246     -0.6979613793
0.6099302143    0.7168252004    -1.276451892
0.6665216775    0.7419769618    -1.282739832
0.6287940354    0.7482649021    -1.307891594
0.5407628704    0.7545528425    -0.9306151724
0.6162181547    0.6728096179    -0.9431910531
0.4904593476    0.7231131407    -0.5910663933
0.3772764212    0.6979613793    -0.3961402423
0.3961402423    0.7734166635    -0.4338678844
0.2703814352    0.8048563653    -0.2829573159
0.2263658527    0.7734166635    -0.1131829264
0.1760623299    0.7734166635    -0.1571985088
0.09431910531   0.7168252004    -0.07545528425
0.06287940354   0.7608407828    -0.02515176142
-0.01886382106  0.6916734389    0.03772764212
-0.1383346878   0.7168252004    0.1509105685
```

```
0.6665216775    0.7419769618    -1.515393625
0.3961402423    0.7042493196    -2.13161178
0.6476578565    0.5407628704    -1.810926822
0.5973543336    0.6790975582    -1.880094166
0.5784905126    0.6979613793    -1.95554945
0.5659146319    0.6979613793    -2.018428854
0.4527317055    0.6350819758    -2.12532384
0.3772764212    0.6476578565    -2.30767411
0.3521246598    0.6413699161    -2.288810289
0.2326537931    0.6979613793    -2.452296738
0.2200779124    0.6476578565    -2.433432917
0.1446226281    0.6099302143    -2.502600261
0.1760623299    0.6602337372    -2.546615843
-0.1006070457   0.5910663933    -2.691238472
-0.1634864492   0.5910663933    -2.747829935
-0.1760623299   0.5596266915    -2.84214904
-0.2640934949   0.5847784529    -2.760405815
-0.2955331966   0.5847784529    -2.8358611
-0.4841714073   0.7168252004    -2.84843698
-0.5847784529   0.622506095     -2.810709338
-0.4778834669   0.6728096179    -2.873588742
-0.5722025722   0.7231131407    -2.829573159
-0.6476578565   0.6728096179    -2.861012861
-0.6350819758   0.6413699161    -2.861012861
-0.6916734389   0.6665216775    -2.898740503
-0.8928875303   0.5847784529    -2.942756086
-0.8488719478   0.6162181547    -2.898740503
-0.8928875303   0.5847784529    -2.942756086
-0.8865995899   0.5784905126    -2.936468145
-1.024934278    0.6099302143    -2.967907847
-1.075237801    0.622506095     -3.030787251
-1.207284548    0.6162181547    -2.961619907
-1.289027773    0.5344749301    -2.898740503
-1.207284548    0.4841714073    -2.961619907
-1.41478658     0.5030352283    -2.8358611
-1.42107452     0.4778834669    -2.873588742
-1.616000671    0.5659146319    -2.760405815
-1.496529804    0.5407628704    -2.861012861
-1.766911239    0.5281869897    -2.67237465
-1.823502703    0.5218990494    -2.697526412
-1.760623299    0.4715955265    -2.678662591
-1.810926822    0.4778834669    -2.716390233
-2.119035899    0.4715955265    -2.540327903
-2.276234408    0.4841714073    -2.200779124
-2.30138617     0.622506095     -2.150475601
```

```
-2.376841454    0.5910663933    -1.873806225
-2.452296738    0.5973543336    -1.804638882
-2.458584678    0.622506095     -1.641152432
-2.48373644     0.5973543336    -1.571985088
-2.464872619    0.5973543336    -1.540545387
-2.389417335    0.5973543336    -1.647440373
-2.464872619    0.5973543336    -1.402210699
-2.433432917    0.6350819758    -1.031222218
-2.420857036    0.6476578565    -1.012358397
-2.471160559    0.622506095     -1.112965443
-2.521464082    0.6728096179    -1.112965443
-2.427144977    0.6413699161    -1.06894986
-2.420857036    0.6665216775    -1.06266192
-2.452296738    0.6476578565    -0.8425840074
-2.357977633    0.6350819758    -0.6916734389
-2.332825871    0.7042493196    -0.622506095
-2.207067064    0.6728096179    -0.5596266915
-2.238506766    0.6916734389    -0.4653075862
-2.169339422    0.7294010811    -0.3961402423
-2.175627362    0.6602337372    -0.3584126002
-2.087596198    0.6162181547    -0.3521246598
-2.018428854    0.7042493196    -0.2075020317
-1.999565033    0.6916734389    -0.2515176142
-1.905245927    0.6728096179    -0.1760623299
-1.94926151     0.7168252004    -0.1634864492
-1.968125331    0.7231131407    -0.1886382106
-1.810926822    0.6413699161    -0.09431910531
-1.735471538    0.6728096179    -0.03772764212
-1.678880075    0.6665216775    0
-1.634864492    0.7482649021    0.08803116496
-1.634864492    0.6979613793    0.1194708667
-1.452514222    0.7042493196    0.09431910531
-1.490241864    0.7168252004    0.07545528425
-1.364483057    0.6476578565    0.1634864492
-1.351907176    0.7482649021    0.1509105685
-1.175844846    0.7168252004    0.2452296738
-1.056373979    0.7419769618    0.2515176142
-0.9872066356   0.7294010811    0.2578055545
-0.9809186952   0.6728096179    0.1886382106
-0.8488719478   0.7294010811    0.2263658527
-0.7734166635   0.7419769618    0.2075020317
-0.6790975582   0.7231131407    0.2200779124
-0.6287940354   0.7294010811    0.194926151
-0.4715955265   0.7608407828    0.2578055545
-0.5722025722   0.7859925442    0.2200779124
```

```
-0.6979613793   0.817432246     0.2515176142
-0.6350819758   0.8677357689    0.2200779124
-0.817432246    0.8928875303    0.3081090773
-0.8865995899   0.8048563653    0.2640934949
-0.8677357689   0.8362960671    0.2075020317
-1.06894986     0.8928875303    0.2766693756
-1.075237801    0.8677357689    0.2075020317
-1.037510158    0.8300081267    0.213789972
-1.144405144    0.8614478285    0.2578055545
-1.213572488    0.8362960671    0.2075020317
-1.24501219     0.8362960671    0.2200779124
-1.326755415    0.7231131407    0.2200779124
-1.41478658     0.7797046039    0.2452296738
-1.452514222    0.7231131407    0.2075020317
-1.559409208    0.6413699161    0.1886382106
-1.609712731    0.622506095     0.2640934949
-1.571985088    0.5659146319    0.194926151
-1.672592134    0.4778834669    0.1760623299
-1.672592134    0.5093231687    0.2200779124
-1.760623299    0.2892452563    0.2578055545
-1.77948712     0.3332608388    0.2640934949
-1.829790643    0.2200779124    0.2829573159
-1.804638882    0.2075020317    0.3081090773
-1.886382106    0.1446226281    0.2452296738
-1.861230345    0.05659146319   0.2640934949
-1.861230345    0.06916734389   0.2389417335
-1.861230345    -0.05659146319  0.2955331966
-1.886382106    -0.06287940354  0.2766693756
-1.924109748    -0.1194708667   0.2200779124
-1.867518285    -0.1571985088   0.2263658527
-1.924109748    -0.1823502703   0.2200779124
-1.917821808    -0.3898523019   0.3081090773
-1.924109748    -0.5910663933   0.2075020317
-1.823502703    -0.6979613793   0.2452296738
-1.735471538    -0.8551598881   0.2829573159
-1.741759478    -0.798568425    0.2452296738
-1.647440373    -0.9620548742   0.2578055545
-1.559409208    -0.9997825163   0.2515176142
-1.540545387    -1.056373979    0.2578055545
-1.546833327    -1.081525741    0.2515176142
-1.446226281    -1.188420727    0.2578055545
-1.395922759    -1.219860429    0.2578055545
-1.257588071    -1.219860429    0.2452296738
-1.326755415    -1.213572488    0.2640934949
-1.200996608    -1.24501219     0.2515176142
```

```
-1.163268965    -1.282739832    0.2766693756
-1.144405144    -1.25130013     0.2766693756
-1.081525741    -1.23872425     0.2578055545
-0.9494789935   -1.263876011    0.2640934949
-0.924327232    -1.320467474    0.3081090773
-0.7734166635   -1.276451892    0.2640934949
-0.8362960671   -1.226148369    0.3269728984
-0.7671287232   -1.23872425     0.2892452563
-0.6162181547   -1.163268965    0.3143970177
-0.4150040634   -1.144405144    0.2955331966
-0.3143970177   -1.106677502    0.2452296738
-0.1760623299   -0.9494789935   0.2955331966
-0.03772764212  -0.7042493196   0.2955331966
-0.1634864492   -0.817432246    0.2200779124
-0.09431910531  -0.7797046039   0.213789972
0.05030352283   -0.6602337372   0.2515176142
0.006287940354  -0.6099302143   0.2640934949
0.04401558248   -0.4401558248   0.2452296738
0.1006070457    -0.1509105685   0.2640934949
0.03143970177   -0.1257588071   0.3521246598
0.03143970177   -0.2766693756   0.213789972
-0.03772764212  0.2515176142    0.2515176142
0.01886382106   0.1006070457    0.2075020317
0.01257588071   0.07545528425   0.2578055545
-0.09431910531  0.3395487791    0.2452296738
0.06287940354   0.04401558248   0.2766693756
0.09431910531   -0.1131829264   0.2452296738
0.06287940354   -0.194926151    0.2452296738
0.05030352283   -0.2200779124   0.2515176142
0       0.1760623299    0.2452296738
-0.03143970177  0.2075020317    0.3081090773
0.006287940354  0.3772764212    0.2515176142
-0.01886382106  0.3332608388    0.2389417335
-0.1006070457   0.496747288     0.2389417335
-0.2578055545   0.5596266915    0.2200779124
-0.3269728984   0.6665216775    0.3206849581
-0.3835643616   0.6728096179    0.2515176142
-0.3772764212   0.6979613793    0.2578055545
-0.3647005405   0.6916734389    0.3584126002
-0.3772764212   0.6665216775    0.2766693756
-0.4464437651   0.7231131407    0.2515176142
-0.5093231687   0.7859925442    0.2829573159
-0.5470508108   0.7608407828    0.2452296738
-0.5659146319   0.7608407828    0.2766693756
-0.1257588071   0.5030352283    0.2578055545
```

```
0        0.3709884809    0.2263658527
-0.07545528425  0.3206849581    0.2452296738
0.006287940354  0.1383346878    0.2640934949
0.1634864492    -0.1006070457   0.2200779124
0.2892452563    -0.1509105685   0.01886382106
0.4275799441    -0.4778834669   -0.06916734389
0.4590196458    -0.4590196458   -0.09431910531
0.3647005405    -0.3458367195   -0.03772764212
0.4527317055    -0.6350819758   -0.2578055545
0.5030352283    -0.7168252004   -0.2766693756
0.5659146319    -0.8614478285   -0.3961402423
0.5659146319    -1.138117204    -0.8614478285
0.5470508108    -1.075237801    -0.8425840074
0.5470508108    -0.9746307549   -0.7231131407
0.5973543336    -1.056373979    -0.7671287232
0.6099302143    -1.119253383    -0.9557669338
0.7545528425    -0.8928875303   -1.584560969
0.7294010811    -0.8865995899   -1.622288611
0.6790975582    -0.9809186952   -1.559409208
0.6916734389    -1.056373979    -1.521681566
0.6099302143    -1.006070457    -1.804638882
0.6099302143    -0.9746307549   -1.974413271
0.5218990494    -0.9494789935   -2.087596198
0.4150040634    -0.9809186952   -2.213355005
0.4715955265    -0.9620548742   -2.251082647
0.4401558248    -0.8803116496   -2.357977633
0.408716123     -0.8614478285   -2.30767411
0.4275799441    -0.9683428145   -2.320249991
0.3898523019    -0.8614478285   -2.389417335
0.3584126002    -0.8928875303   -2.420857036
0.2703814352    -0.817432246    -2.49002438
0.2200779124    -0.7168252004   -2.603207307
0.194926151     -0.6602337372   -2.691238472
0.03143970177   -0.4212920037   -2.804421398
0.09431910531   -0.3709884809   -2.760405815
0.01886382106   -0.1823502703   -2.829573159
-0.03143970177  0.194926151     -2.804421398
-0.3898523019   0.5847784529    -2.760405815
-0.6099302143   0.9117513513    -2.603207307
-0.817432246    1.119253383     -2.433432917
-0.8362960671   1.112965443     -2.427144977
-0.8991754706   1.276451892     -2.181915303
-0.9306151724   1.553121267     -1.917821808
-0.9809186952   1.60342479      -1.534257446
-0.9306151724   1.622288611     -1.559409208
```

```
-0.9809186952   1.60342479      -1.502817745
-0.9117513513   1.710319776     -1.232436309
-0.9306151724   1.622288611     -1.207284548
-0.8551598881   1.685168015     -1.452514222
-0.8551598881   1.584560969     -1.471378043
-0.9494789935   1.584560969     -1.377058938
-0.9431910531   1.59713685      -1.351907176
-0.8991754706   1.634864492     -1.270163952
-0.8677357689   1.622288611     -1.175844846
-0.9306151724   1.584560969     -1.131829264
-0.8803116496   1.609712731     -1.094101622
-0.9180392917   1.653728313     -1.024934278
-0.8551598881   1.634864492     -0.9620548742
-0.924327232    1.660016253     -0.905463411
-0.8991754706   1.634864492     -0.7922804846
-0.9180392917   1.584560969     -0.6539457968
-0.8237201864   1.571985088     -0.622506095
-0.8300081267   1.515393625     -0.4464437651
-0.8237201864   1.458802162     -0.3584126002
-0.8865995899   1.458802162     -0.408716123
-0.3269728984   0.9934945759    0.07545528425
-0.3898523019   1.006070457     0.05659146319
-0.3521246598   0.7545528425    0.2075020317
-0.3332608388   0.7545528425    0.2640934949
-0.3835643616   0.4275799441    0.3898523019
-0.1760623299   1.131829264     -0.09431910531
0.01257588071   1.075237801     -0.4275799441
-0.01886382106  1.112965443     -0.3772764212
0       1.23872425      -0.6287940354
0       1.219860429     -0.5973543336
0.05030352283   1.307891594     -0.8551598881
0       1.301603653     -0.8300081267
0.05030352283   1.339331295     -0.9620548742
0.09431910531   1.333043355     -0.9997825163
0.1131829264    1.276451892     -1.23872425
0.1886382106    1.200996608     -1.182132787
0.09431910531   1.351907176     -1.490241864
0.1006070457    1.276451892     -1.748047418
0.1886382106    1.024934278     -2.062444436
0.1320467474    0.8362960671    -2.269946468
0.1383346878    0.8614478285    -2.219642945
0.2263658527    0.5281869897    -2.427144977
0.1760623299    0.5407628704    -2.408281156
0.1446226281    0.2515176142    -2.628359068
0.06287940354   0.07545528425   -2.741541994
```

```
-0.4275799441   -0.1697743896   -3.055939012
-0.603642274    -0.7042493196   -3.087378714
-0.6728096179   -0.6790975582   -3.049651072
-0.6665216775   -0.8048563653   -3.043363131
-0.7671287232   -1.024934278    -2.961619907
-0.798568425    -1.043798099    -2.911316384
-1.018646337    -1.641152432    -2.647222889
-0.9557669338   -1.804638882    -2.571767605
-1.169556906    -1.930397689    -2.288810289
-1.213572488    -1.930397689    -2.257370587
-1.213572488    -2.012140913    -2.257370587
-1.333043355    -2.068732376    -1.999565033
-1.276451892    -2.043580615    -2.012140913
-1.351907176    -2.181915303    -1.584560969
-1.307891594    -2.207067064    -1.622288611
-1.553121267    -2.037292675    -1.370770997
-1.622288611    -2.106460019    -1.395922759
-1.509105685    -2.13161178     -1.094101622
-1.351907176    -2.018428854    -0.5659146319
-1.194708667    -1.886382106    -0.2892452563
-1.232436309    -1.861230345    -0.3269728984
-2.30138617     -1.458802162    -0.8865995899
-2.339113812    -1.465090102    -0.817432246
-2.232218826    -1.660016253    -1.213572488
-2.081308257    -1.817214762    -1.641152432
-2.087596198    -1.792063001    -1.634864492
-1.980701212    -1.741759478    -1.848654464
-2.056156496    -1.760623299    -1.031222218
-2.295098229    -1.433650401    -0.798568425
-2.263658527    -1.333043355    -0.4464437651
-2.232218826    -1.282739832    -0.4150040634
-2.087596198    -1.200996608    -0.2452296738
-2.018428854    -0.9494789935   0.1131829264
-1.905245927    -0.7734166635   0.1760623299
-1.854942404    -0.798568425    0.1823502703
-1.766911239    -0.6099302143   0.3584126002
-1.678880075    -0.6476578565   0.3081090773
-1.924109748    -0.5093231687   0.1886382106
-1.861230345    -0.496747288    0.2955331966
-1.905245927    -0.4150040634   0.2200779124
-1.854942404    -0.4527317055   0.2766693756
-1.842366524    -0.5093231687   0.2955331966
-1.867518285    -0.3081090773   0.3332608388
-1.836078583    -0.3395487791   0.3206849581
-1.653728313    -0.2640934949   0.3584126002
```

```
-1.672592134    -0.2640934949   0.4527317055
-1.60342479     -0.1571985088   0.4778834669
-1.521681566    -0.1571985088   0.4464437651
-1.263876011    0.08803116496   0.4841714073
-1.263876011    0.05659146319   0.5470508108
-1.23872425     0.1131829264    0.4590196458
-1.685168015    0.1697743896    0.4338678844
-1.169556906    0.4590196458    0.3772764212
-1.169556906    0.4590196458    0.4338678844
-0.8865995899   0.622506095     0.3898523019
-0.9809186952   0.6728096179    0.3898523019
-0.9620548742   0.5910663933    0.3772764212
-1.138117204    0.5281869897    0.4338678844
-1.138117204    0.4590196458    0.4212920037
-0.8048563653   0.8048563653    0.3458367195
-0.8677357689   0.7734166635    0.2829573159
-0.7734166635   0.7859925442    0.3143970177
-0.5344749301   0.5973543336    0.3206849581];

%% ----------------  [Magneto Data]  ---------------- %%

A = [0.023431 , -0.000513 , -0.000139;
    -0.000513 , 0.021650 , -0.000275;
    -0.000139 , -0.000275 , 0.022589];

B = [-22.141157 , -6.868126 , -30.821610];

D = (A * (C - B)')';

MagD = D(:,1).^2 + D(:,2).^2 + D(:,3).^2;

%% --------------------  [Plot]  -------------------- %%

figure;
hold on;
grid on;
p1 = plot3(C1(:,1),C1(:,2),C1(:,3),'.','MarkerSize', 15);
p2 = plot3(D(:,1),D(:,2),D(:,3),'.','MarkerSize', 15);
ellipsoid(C1(:,1),C1(:,2),C1(:,3),[0 0.4470 0.7410],0.1);
ellipsoid(D(:,1),D(:,2),D(:,3),[0.8500 0.3250 0.0980],0.1);
legend('Uncalibrated Magnetometer Data','Calibrated
   Magnetometer Data');
xlabel('X'), ylabel('Y'), zlabel('Z');
title('Magnetometer Data 3D Scatterplot');
view(45,10);
```

```
axis vis3d;

function ellipsoid(x,y,z,color,trans)

% Matlab Add-on for Ellipsoid Fit by Yury %
%<https://www.mathworks.com/matlabcentral/fileexchange/24693-
   ellipsoid-fit>
[ center, radii, evec, v, chi ] = ellipsoid_fit( [ x y z ], ''
   );

size_min = min( [ x y z ] );
size_max = max( [ x y z ] );

steps = 100;
step = ( size_max - size_min ) / steps;

[ x, y, z ] = meshgrid( linspace( size_min(1) - step(1),
   size_max(1) + step(1), steps ), linspace( size_min(2) - step
   (2), size_max(2) + step(2), steps ), linspace( size_min(3) -
    step(3), size_max(3) + step(3), steps ) );

E = v(1) *x.*x +   v(2) * y.*y + v(3) * z.*z + ...
          2*v(4) *x.*y + 2*v(5)*x.*z + 2*v(6) * y.*z + ...
          2*v(7) *x    + 2*v(8)*y    + 2*v(9) * z;
e = patch( isosurface( x, y, z, E, -v(10) ) );
set( e, 'FaceColor',color, 'EdgeColor', 'none','FaceAlpha',
   trans);
end
```

Appendix L — Simulink MATLAB Function: Reduced IMU Calibration Code for Yaw using Gyroscope Accelerometer

```
function [calibrated_yaw,calibrated_rate,ready]= EularAngle(rate,mag)

% ---------------- [Tunable User Variables] ---------------- %

n = 30; % Max Cycles for Calibrating IMU/MPU data

% ------------------- [Function Counter] ------------------- %
persistent callCount loopcycle;

if isempty(loopcycle), loopcycle = 0; end                % If caibration is done, then loop cycle = 1

    if isempty(callCount)
        callCount = 0; % Initialize on first call
    end
    callCount = callCount + 1; % +1 for each time the function is called

% ------------------- [Function Counter] ------------------- %

A = [0.025257 , 0.000062 , -0.000096;
    0.000062 , 0.023110 , -0.000343;
    -0.000096 , -0.000343 , 0.024714];

B = [-27.385238 , 6.142666 , -33.779326];

cal_Mag = A* (mag - B)';

Mag_yaw = rad2deg(atan2(cal_Mag(2,1),cal_Mag(1,1)));

% ------------ [Defining Persistent Variables] ------------ %
persistent average_rate offset_rate cal_rate;

if isempty(average_rate), average_rate = [0;0;0]; end    % Average Gyro Rate
if isempty(offset_rate), offset_rate = [0;0;0]; end      % Gyro Offset
if isempty(cal_rate), cal_rate = [0;0;0]; end            % Calibrated Gyro Rate
```

```
35
36 if callCount < n
37     average_rate = average_rate + rate';
38 else
39
40     offset_rate = average_rate/n;
41     cal_rate = rate' - offset_rate;
42     loopcycle = 1;
43 end
44
45 % ----------------------- [Return] ----------------------- %
46
47 calibrated_rate = rad2deg(cal_rate(3,1));
48 calibrated_yaw = Mag_yaw;
49 ready = loopcycle;
```

Appendix M — Simulink MATLAB Function: Reduced IMU Complementary Filter for Yaw

```
function eug_yaw = CompFilter(gyro,mag,timestep)

% ------------ [Defining Persistent Variables] ------------ %
persistent yaw;

if isempty(yaw), yaw = [0]; end

% ---------------- [Tunable User Variables] ---------------- %
alpha = 0.98;

% ----------------- [Complementary Filter] ----------------- %
yaw = alpha*(gyro(1,1)*timestep + yaw) + (1-alpha)*(mag(1,1));
eug_yaw = yaw;
```