



**Near Real-Time Closed-Loop Optimal Control Feedback for Spacecraft
Attitude Maneuvers**

THESIS

C. Douglas McFarland, Captain, USAF

AFIT/GA/ENY/09-M06

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GA/ENY/09-M06

NEAR REAL-TIME CLOSED-LOOP OPTIMAL CONTROL FEEDBACK FOR
SPACECRAFT ATTITUDE MANEUVERS

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

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

C. Douglas McFarland, BS

Captain, USAF

March 2009

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

NEAR REAL-TIME CLOSED-LOOP OPTIMAL CONTROL FEEDBACK FOR
SPACECRAFT ATTITUDE MANEUVERS

C. Douglas McFarland, BS
Captain, USAF

Approved:

<hr/>	<hr/>
//signed//	16 Mar 2009
Lt Col Eric Swenson (Chairman)	Date
<hr/>	<hr/>
//signed//	16 Mar 2009
Dr. Richard Cobb (Member)	Date
<hr/>	<hr/>
//signed//	16 Mar 2009
Dr. Jonathan Black (Member)	Date
<hr/>	<hr/>
//signed//	16 Mar 2009
Lt Col Frederick Harmon (Member)	Date

Abstract

Optimization of spacecraft attitude maneuvers can significantly reduce attitude control system size and mass, and extend satellite end-of-life. Optimal control theory has been applied to solve a variety of open-loop optimal control problems for terrestrial, air, and space applications. However, general application of real-time optimal controllers on spacecraft for large slew maneuvers has been limited because open-loop control systems are inherently vulnerable to error and the computation necessary to solve for an optimized control solution is resource intensive. This research effort is focused on developing a near real-time optimal control (RTOC) system for spacecraft attitude maneuvers on the Air Force Institute of Technology's 2nd generation simulated satellite, SimSat II. To meet the end goal of developing a RTOC controller, necessary preliminary steps were completed to accurately characterize SimSAT II's mass properties and attitude control system. Using DIDO, a pseudospectral-based optimal control solver package, to continuously solve and execute a sequence of optimized open-loop control solutions in near real-time, the RTOC controller can optimally control the state of the satellite over the course of a large angle slew maneuver. In this research, simulation and experimental results clearly demonstrate the benefit of RTOC versus other non-optimal control methods for the same maneuver.

Acknowledgements

This thesis was completed solely because a great number of people were able to put up with my excessive pestering, badgering, and innate ability to break hardware. To Lt Col Swenson, thank you for your persistent dedication to helping me see this project through. Without your guidance, assistance, and reassurance this research would not have been possible. To Jay, Wilbur, Chris and the other lab technicians who assisted me with SimSAT II, thank you for your guidance with machining and electronics, always being able to get the hardware I needed, and always being able to lend a hand at a moments notice. Believe it or not, I also appreciated your constant reminders to be safety vigilant. I still have all ten fingers and toes, so I consider your mission a success. To Dr. Ross and Dr. Sekhavat, thank you for taking the time to personally help me with implementing a DIDO-based RTOC controller on SimSAT II and for providing me with excellent guidance for debugging my code. To Dr. Robbie Robertson at AFRL/RV, thank you for taking the opportunity to sponsor this research effort. To my friends, thank your for putting up with me, even as grumpy as I was at times.

C. Douglas McFarland

Table of Contents

	Page
Abstract	iv
Acknowledgements	v
List of Figures	viii
List of Tables	xi
I. Introduction	1
1.1 Background	1
1.2 Research Objectives	2
1.3 Methodology	3
1.4 Thesis Outline	4
II. Background	5
2.1 Spacecraft Dynamics	5
2.1.1 Center of Mass	7
2.1.2 Angular Momentum and Mass Moment of Inertia	7
2.1.3 Euler Rotational Equations	10
2.1.4 Rotational Kinematics	12
2.1.5 Quaternion Parameters	15
2.2 Satellite Simulators	19
2.2.1 Historical Simulated Satellites	20
2.2.2 AFIT SimSATs	21
2.3 System Characterization	22
2.3.1 Mass Properties Determination	22
2.3.2 Thruster Characterization	25
2.4 Optimal Control Theory	28
2.4.1 Fixed Initial Time, Free Final Time Optimal Control	30
2.4.2 Pseudospectral Methods	33
2.4.3 Real-Time Optimal Control	34
2.4.4 Space Application of Optimal Control	35
2.5 Summary	36
III. System Architecture and Characterization	37
3.1 Hardware	37
3.1.1 Ground Station	37
3.1.2 Air Bearing	39
3.1.3 SimSAT II Hardware	39
3.2 Software	45

	Page
3.2.1	49
3.3	50
3.4	51
3.4.1	52
3.4.2	57
3.4.3	60
3.5	68
3.5.1	71
3.5.2	71
3.6	72
IV.	74
4.1	74
4.2	79
4.3	80
4.4	85
V.	86
5.1	86
5.2	89
5.2.1	90
5.2.2	94
5.3	101
5.3.1	106
5.4	113
5.4.1	114
5.4.2	117
5.4.3	120
5.4.4	122
5.4.5	123
VI.	129
6.1	129
6.2	130
Bibliography	133

List of Figures

Figure		Page
1.1	SimSAT II Simulated Satellite Testbed	3
2.1	Angular Momentum of a Particle	7
2.2	Frame of Reference Tied to a Rigid Body	8
2.3	Standard Aerospace 1-2-3 Euler Angle Set	13
2.4	1-2-3 Euler Angle Rotation from Inertial Frame to Body Frame	14
2.5	Rotation about a Direction Vector Shared by Reference Frames	16
2.6	AEROTECH AHL9000 Series Planar Air Bearing [3]	20
2.7	“Tabletop” and “Umbrella” Spherical Air Bearing Configurations	21
2.8	“Dumbbell” Spherical Air Bearing Configuration	22
2.9	AFIT’s SimSAT I	23
2.10	Misalignment of CG and CR Causing Torque on a Rigid Body	24
2.11	Idealized Propeller Thrust Production	26
2.12	Example of Bellman’s Principle of Optimality	34
2.13	Real-Time Optimal Control - Control Loop	35
3.1	AFIT’s SimSAT II simulated satellite system	38
3.2	Space Electronics, Inc. Tri-Axis Spherical Air Bearing	40
3.3	SimSAT II Top (Left) and Bottom (Right) View	41
3.4	SimSAT II Mini-box PC	42
3.5	Northrup Grumman LN-200 Fiber Optic Gyroscope	44
3.6	DS1401/1501 dSPACE MicroAutoBox	45
3.7	#118895 Maxon EC Motor and Landing Products LP05050 5x5 Propeller	46
3.8	SimSAT II Top-level Simulink Controller Model	47
3.9	Example ControlDesk Layout for SimSAT II	48
3.10	SimSAT II Hardware Architecture	50
3.11	SimSAT II Software Architecture	51
3.12	RS-232 Blocks in SimSAT II Simulink Control Model	52
3.13	Example LN-200 Gyro Data Spike	53
3.14	Bode Plot of a Butterworth Low-Pass Filter [2]	54
3.15	Digital Filter and Rate Limiting Techniques on Isolated Gyro Corruption	55
3.16	Effects of Over-Constrained Rate Limiting of Oscillating Signal	56
3.17	Rate Limiter Blocks in SimSAT II Simulink Control Model	57

	Page
3.18	Initial MOI Estimation Using Fan Thruster Torque 59
3.19	Error in Known Torque due to Cosine Losses 60
3.20	Static Thrust Characterization Setup 62
3.21	Discrete Derivative of Measured Angular Rate 65
3.22	Angular Acceleration Slope Intervals for 3-Axis Thrusters at 2500 RPM . 67
3.23	Comparison of Thruster Characterization Results 69
3.24	Top-Level View of SimSAT II Control Model 70
3.25	“mlib” Function Command Delay 72
4.1	Implementation of OLOC Controller on SimSAT II 80
4.2	RTOC Control Loops 81
4.3	Implementation of RTOC Control on SimSAT II 82
4.4	Implementation of RTOC Control on SimSAT II 83
4.5	RTOC Controller Logic 85
5.1	SimSAT II Body-Frame Axes 87
5.2	SimSAT II Simulation Results - OLOC Controller 45° Maneuver 91
5.3	SimSAT II Experimental Results - OLOC Controller 45° Maneuver 93
5.4	SimSAT II Simulation Results - OLOC Controller 90° Maneuver 95
5.5	SimSAT II Experimental Results - OLOC Controller 90° Maneuver 96
5.6	SimSAT II Simulation Results - OLOC Controller 180° Maneuver 97
5.7	SimSAT II Experimental Results - OLOC Controller 180° Maneuver 98
5.8	SimSAT II Simulation Results - EROC Controller 45° Maneuver 99
5.9	SimSAT II Experimental Results - EROC Controller 45° Maneuver 100
5.10	SimSAT II Simulation Results - EROC Controller 90° Maneuver 102
5.11	SimSAT II Experimental Results - EROC Controller 90° Maneuver 103
5.12	SimSAT II Simulation Results - EROC Controller 180° Maneuver 104
5.13	SimSAT II Experimental Results - EROC Controller 180° Maneuver 105
5.14	SimSAT II Model States - Real-time Update RTOC Controller 45° Maneuver 107
5.15	ϵ_{inner} - Real-time Update RTOC Controller 45° Maneuver 108
5.16	SimSAT II Simulation - Real-time Update RTOC Controller 45° Ma- neuver Final Time 109
5.17	ϵ_{outer} - Real-time Update RTOC Controller 45° Maneuver 110
5.18	SimSAT II Simulation - Real-time Update RTOC Controller 90° Maneuver 111
5.19	SimSAT II Simulation - Real-time Update RTOC Controller 180° Maneuver 112

		Page
5.20	Quaternion States - OLOC Controller Maneuvers Using Quaternion Boundary Constraints	113
5.21	Angular Rate Simulation - LGL Node Variation for OLOC Controller 45° Maneuver	115
5.22	Control Simulation - LGL Node Variation for OLOC Controller 45° Maneuver	116
5.23	Angular Rate - Update Time Variation for RTOC Controller 45° Maneuver	118
5.24	DIDO Computation Time - 45° Real-time Update RTOC Maneuver	119
5.25	Different Optimal Solutions - Real-time Update RTOC Controller 45° Maneuver	120
5.26	DIDO Optimal Control Solution Output - Hamiltonian H	121
5.27	Angular Rotation Rate Simulation - Solution Guess Variation for 1s Update RTOC Controller 45° Maneuver	123
5.28	Maneuver Final Time - Solution Guess Variation for 1s Update RTOC Controller 45° Maneuver	124
5.29	SimSAT II Simulation - 10% MOI Variation for Real-time Update RTOC Controller 180° Maneuver	125
5.30	SimSAT II Simulation - 10% MOI Variation for Real-time Update RTOC Controller 180° Maneuver	126
5.31	ϵ_{inner} - 10% MOI Variation for Real-time Update RTOC Controller 180° Maneuver	127
5.32	Maneuver Final Time - 10% MOI Variation for Real-time Update RTOC Controller 180° Maneuver	128

List of Tables

Table		Page
3.1	SimSAT II Ground Station PC Components	39
3.2	Space Electronics, Inc. Tri-Axis Air Bearing	39
3.3	SimSAT II Mini-box PC Components	42
3.4	Northrop Grumman LN-200 IMU	43
3.5	DS1401/1501 dSPACE MicroAutoBox	44
3.6	SimSAT II MOI Matrix - Fully Loaded	57
3.7	Final SimSAT Measured MOI Values	60
3.8	Positive ω_i Static Thrust Fan Characterization Polynomial Coefficients . .	62
3.9	Negative ω_i Static Thrust Fan Characterization Polynomial Coefficients .	62
4.1	Coefficients for SimSAT II's Air Drag Polynomial Function	78
5.1	OLOC Simulation and Experimental Testing Parameters	90
5.2	Simulation Results - OLOC Performance Indices	94
5.3	OLOC and EROC Controller Simulation Performance Index Comparison	101
5.4	RTOC Controller Simulation Testing Parameters	101
5.5	Optimal Control Performance Index Comparison	109
5.6	DIDO Node Versus Performance Index Comparison	114
5.7	DIDO Node Versus Solution Calculation Time Comparison	117

NEAR REAL-TIME CLOSED-LOOP OPTIMAL CONTROL FEEDBACK FOR SPACECRAFT ATTITUDE MANEUVERS

I. Introduction

1.1 Background

Optimal control theory (OCT) has been employed in the past to solve a variety of open-loop optimal control problems for terrestrial, air, and space applications. Every day we unwittingly apply OCT on the way home from work each night. Naturally, one chooses the fastest route home, however, getting home in the least amount of time does not necessarily mean following the shortest distance path from work to home. A traffic accident may have closed down the main highway, so taking the longer back-roads route may be the minimum time route. OCT concepts can be applied to a convoy heading from one base camp to another, a UAV loitering above its target, or a spacecraft changing its attitude. By applying OCT techniques to the physical equations governing the systems, engineers can put metrics on savings that can clearly be realized in system design.

A recent example of OCT being exercised in a space application is the Zero-Propellant Maneuver performed by the International Space Station (ISS). Researchers at Draper Laboratories, Rice University, and the Naval Post Graduate School applied OCT to reorient the ISS to a new attitude without the use of any propellant. Previously, the ISS's Control-Moment Gyroscopes (CMG) would reach their rate limit in the middle of the reorientation. As a result, the ISS required the use of its cold-gas thrusters to complete the full reorientation maneuver. Researchers using OCT identified that naturally occurring aerodynamic drag and gravity forces could be exploited in such a way that the CMGs could complete the reorientation maneuver without reaching their rate limit. By removing the need for using the ISS's cold-gas thrusters for attitude reorientation, OCT effectively reduced the time between ISS refueling. In the end, NASA eliminated considerably refueling costs as it enters a critical juncture in its history with the retirement of the Space Shuttle [7].

General use of OCT for large angle spacecraft slew maneuvers has been limited by an open-loop control systems' inherent vulnerability to external disturbances and discrepancies between the physical plant and model. In this research, OCT is used to compute an open-loop control solution for minimum time spacecraft slew maneuvers. If a spacecraft experiences any disturbance torques or differences between model and actual parameters, then the computed optimal control solution for a given maneuver may not really be optimal. For example, a solar flare can add to the drag induced on a LEO spacecraft by the Earth's atmosphere, resulting in the system reacting differently to control inputs than under normal drag conditions. At best, these changes in performance may result in a lack of optimality in terms of a small error in the end state of a maneuver. At worst, it may result in complete loss of the satellite.

To combat the problems caused by events affecting a spacecraft's dynamics, it is possible to continuously recompute the optimal control solution during the course of a maneuver. By closing the control loop, the spacecraft always keeps following an optimal control trajectory solution even in the presence of disturbances and model discrepancies. Calculating an optimal control solution for a spacecraft reorientation maneuver is computationally intensive because nonlinear programming techniques must be used to solve the coupled, highly non-linear differential equations of motion of a spacecraft. Presently, current computational capabilities and software implementation limitations prevent the calculation of optimal control solutions from being fast enough to be continuously recomputed by a spacecraft during a reorientation maneuver. However, given the rate that computers are growing in computational capabilities, real-time optimal control will likely be realized in the near future.

1.2 Research Objectives

This research is focused on developing, evaluating, and testing a near real-time optimal control (RTOC) spacecraft attitude controller for the AFIT's 2nd generation ground-based simulated satellite testbed, SimSAT II, pictured in Figure 1.1. The RTOC controller closes the optimal control loop while SimSAT II is performing an attitude

reorientation maneuver, effectively allowing SimSAT II to maintain an optimal control trajectory while experiencing disturbance torques or uncertainty in system parameters.

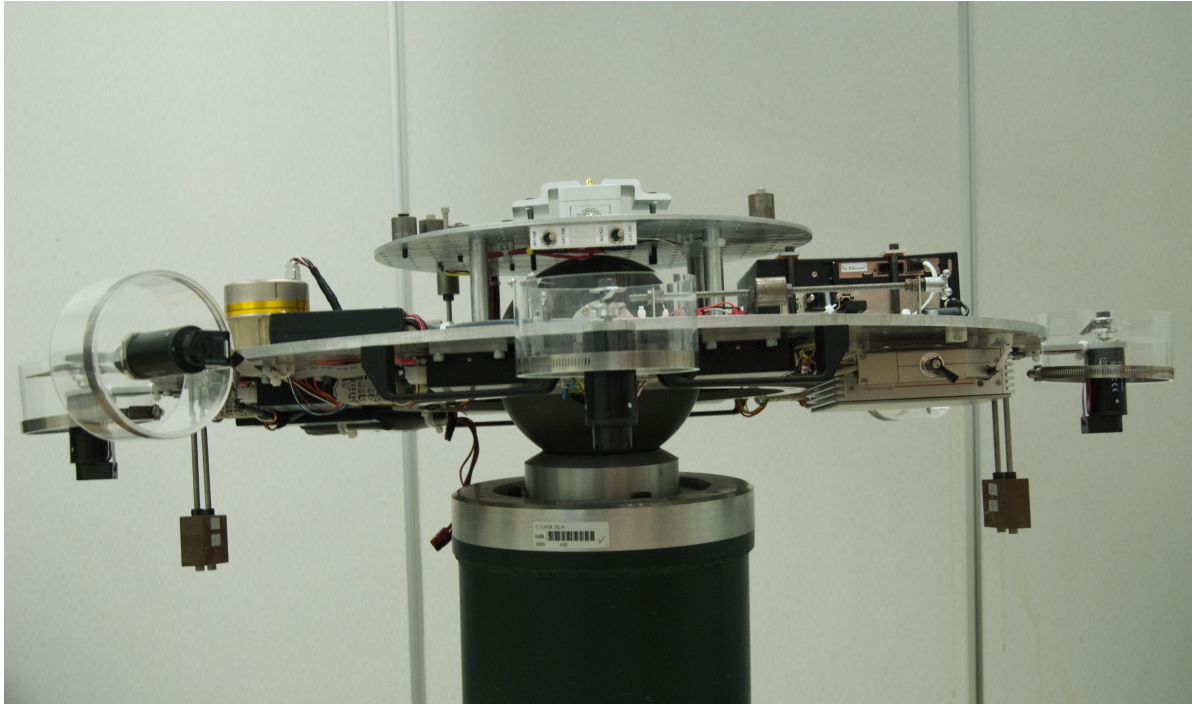


Figure 1.1: SimSAT II Simulated Satellite Testbed

1.3 Methodology

This research began by first simulating the open-loop optimal control (OLOC) controller model in MATLAB. The OLOC controller uses the pseudospectral (PS) based optimal control solving program called DIDO to solve the coupled and highly nonlinear equations of motion of a spacecraft, which cannot be solved analytically. The DIDO application has demonstrated the ability to solve optimal control problems in as little as a few seconds, which for the purposes of this research will be considered near real-time. The RTOC controller is able to close the optimal control loop by immediately executing control solutions either in a model or on-board SimSAT II and sending the resulting state information back to the OLOC controller to recalculate a new optimal control solution. Once the OLOC and RTOC controllers can simulate SimSAT II performing an attitude reorientation maneuver, the OLOC and RTOC controllers will be implemented into the SimSAT II testbed.

Before an optimal controller could be implemented on SimSAT II, several preliminary hardware integration and characterization steps were completed. First, a LN-200 Fiber-optic Gyroscope (FOG) was integrated with SimSAT II's existing hardware and software architecture to enable SimSAT II to measure its angular position and rotation rates. Next, SimSAT II's mass moment of inertia properties were determined to accurately calculate an optimal control solution. And lastly, the torque produced by SimSAT II's simulated thrusters was characterized in order to properly implement the calculated optimal control solutions.

1.4 Thesis Outline

Chapter II provides an extensive literature review on spacecraft dynamics, satellite simulators, hardware characterization, and OCT. Chapter III describes SimSAT II's hardware and software, and methods and results of preliminary hardware integration and characterization. Chapter IV describes the methods used for simulating and physically implementing optimal control. Chapter V provides the results and analysis of optimal control simulation and experimental testing on SimSAT II. Chapter VI provides conclusions and recommendations for future research.

II. Background

THIS chapter provides a literature review of topics relevant to this research. The first section reviews spacecraft dynamics, followed by a historical overview of satellite simulators. The next section discusses spacecraft characterization methods. The last section explains optimal control theory (OCT) and how it has been used in related research.

2.1 *Spacecraft Dynamics*

This section provides an overview of attitude dynamics for a spacecraft. This discussion starts with the concept of rigid body dynamics. The foundation for rigid body dynamics is the concept of particle dynamics as described by Isaac Newton in his *Principia Mathematica Philosophia Naturalis* circa 1687. Wiesel concisely summarized Newton's concept for three laws of particle motion as [54]:

1. A particle at rest remains at rest, and a particle in motion remains in motion, if the applied force is zero.
2. The force on a particle equals the mass of the particle times its inertial acceleration.
3. For every applied force, there is an equal and opposite reaction force.

These three laws, along with Newton's law of gravity, constitute a complete description of the laws of mechanics. In modern vector notations, the second law takes the form

$$\mathbf{F} = m\mathbf{a} \tag{2.1}$$

where \mathbf{F} is the total force on the particle and \mathbf{a} is its acceleration with respect to the inertial frame.

A particle can be described as a body whose entire mass m acts at a single point. When an external force acts on a particle in free space, it only has three translational degrees of freedom (DOF).

A rigid body can be described as a system of particles, where the geometric distance between any two particles in the body remains constant [52]. When an external force acts on a rigid body, the external force is said to act on one of the many constituent

particles that comprises that rigid body. Since the geometric position between any two constituent particles is constant, if one of the particles accelerates, with the exception of pure rotation of a rigid body, then all of the particles that comprise the rigid body must also accelerate. For the pure rotation case, particles located exactly on the axis of rotation do not accelerate because they are only rotating. Exceptions aside, if a particle accelerates, then a force must be associated with acceleration. Therefore, there must be internal forces to the rigid body that are acting on each of the rigid body's constituent particles. Equation (2.2) redefines Eq. (2.1) as a rigid body comprised of constituent particles in the form of

$$\mathbf{F}_i = \mathbf{f}_{ie} + \sum_{i \neq j}^N \mathbf{f}_{ij} = m_i \mathbf{a}_i \quad (2.2)$$

where \mathbf{F}_i is the total force acting on particle i , \mathbf{f}_{ie} is the external force acting on particle i , \mathbf{f}_{ij} are the internal rigid body forces from particle i acting on the j th constituent particle, and $m_i \mathbf{a}_i$ is the resultant mass and acceleration of particle i . When the number of constituent particles in a rigid body is four or greater, a rigid body has six DOF [54].

If a rigid body is made of j constituent particles, there must be j force equations of the form expressed in Eq. (2.2), one for each particle. The total sum results in

$$\sum_{i=1}^N \mathbf{f}_{ie} + \sum_{i=1}^N \sum_{i \neq j}^N \mathbf{f}_{ij} = \sum_{i=1}^N m_i \mathbf{a}_i. \quad (2.3)$$

Applying Newton's third law, the summation of the internal forces \mathbf{f}_{ij} must be zero [54]. Therefore, Eq. (2.3) can be reduced to

$$\mathbf{F}_e = \sum_{i=1}^N \mathbf{f}_{ie} \quad (2.4)$$

where \mathbf{F}_e is the total or resultant external force acting on the i -th rigid body. As the number of i constituent particles approaches a large number, the summation in Eq. (2.4) can be expressed as an integral over the entire rigid body written as

$$\mathbf{F}_e = \int_{body} d\mathbf{f}_{ie} \quad (2.5)$$

where $d\mathbf{f}_{ie}$ is a differential force on an infinitely small constituent particle [54].

2.1.1 Center of Mass. The center of mass of a rigid body is defined by

$$\mathbf{r}_c = \frac{1}{M} \sum_{i=1}^N m_i \mathbf{r}_i \quad (2.6)$$

where \mathbf{r}_c is vector pointing from an arbitrary reference point to the center of mass of the rigid body, M is the total mass, and $m_i \mathbf{r}_i$ is the mass and location of the i th constituent particle with respect to an arbitrary reference point. By assuming an infinite number of constituent particles, the summation in Eq. (2.6) can be replaced with an integral throughout the entire body

$$\mathbf{r}_c = \frac{1}{M} \int_{body} \mathbf{r}_i dm \quad (2.7)$$

where dm is the mass of a small volume [54]. Therefore, Eq. (2.3) can now be defined in terms of

$$\mathbf{F}_e = M \frac{d^2}{dt^2}(\mathbf{r}_c) \quad (2.8)$$

where $\frac{d^2}{dt^2}(\mathbf{r}_c)$ is the inertial acceleration of the rigid body's center of mass.

2.1.2 Angular Momentum and Mass Moment of Inertia. The angular momentum of a particle is defined by

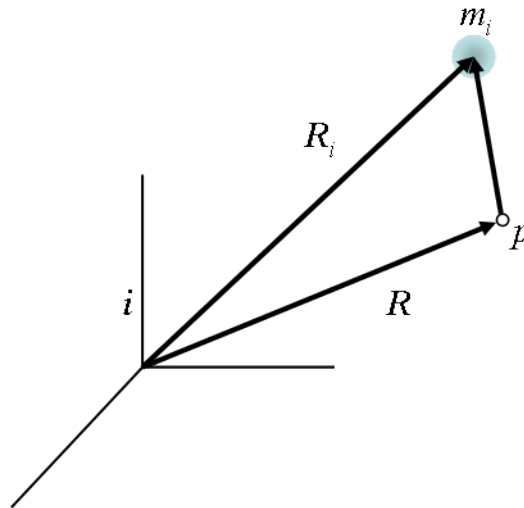


Figure 2.1: Angular Momentum of a Particle

$$\mathbf{H} = m\mathbf{r} \times \mathbf{v} \quad (2.9)$$

where \mathbf{H} is the angular momentum of the particle, as shown in Figure 2.1, about a desired point, m is the mass of the particle, \mathbf{r} is a position vector from where the angular momentum is being found to the location of the particle, and \mathbf{v} is the velocity vector of the particle tangential to its position vector \mathbf{r} .

The angular momentum for a rigid body can similarly be found by

$$\mathbf{H} = \int_{body} \mathbf{r} \times \mathbf{v} dm \quad (2.10)$$

where the vector quantity for the angular momentum in the body frame coordinate system is \mathbf{H} . This is done in order to simplify the mathematics, because the position vector \mathbf{r} for each mass element remains constant with respect to the body frame and can be thought of as tied to the body, as shown in Figure 2.2 [54]. If the position vector \mathbf{r} is expressed in the inertial coordinate frame, it would constantly be changing, making the calculation of angular momentum much more complicated. The inertial derivative of the position vector \mathbf{r} is the inertial velocity of each particle about the center of mass. For each mass element, an inertial derivative of the position vector would also need to be computed to obtain a velocity vector expressed in body frame coordinates as

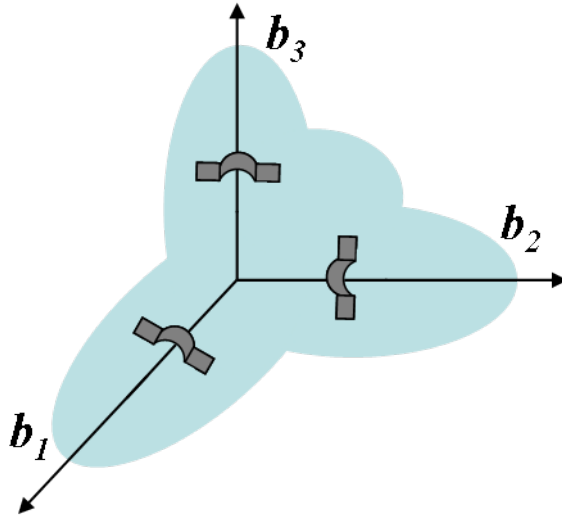


Figure 2.2: Frame of Reference Tied to a Rigid Body

$$\frac{d}{dt}\mathbf{r} = \dot{\mathbf{r}} + \omega^{bi} \times \mathbf{r}. \quad (2.11)$$

Since the position of each mass element is constant with respect to the body frame for a rigid body, any changes in the velocity vector are due only to its rotation about the body frame ω^{bi} . Therefore, $\dot{\mathbf{r}}$ in Eq. (2.11) is zero.

Substituting Eq. (2.11) into Eq. (2.10) results in

$$\mathbf{H} = \int_{body} \mathbf{r} \times (\omega^{bi} \times \mathbf{r}) dm. \quad (2.12)$$

The position vector \mathbf{r} and angular velocity vector ω^{bi} in Eq.(2.12) expressed in component form in the body coordinate frame is

$$\mathbf{r} = x\mathbf{b}_1 + y\mathbf{b}_2 + z\mathbf{b}_3 \quad (2.13)$$

and

$$\omega^{bi} = \omega_1\mathbf{b}_1 + \omega_2\mathbf{b}_2 + \omega_3\mathbf{b}_3 \quad (2.14)$$

where $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$ represent the body frame coordinate system direction vectors. The scalar components of the position vector are x , y , and z and the scalar angular velocity components of the body frame coordinate systems rotation with respect to the inertial coordinate frame are ω_1 , ω_2 , and ω_3 . Using the vector multiplication identity

$$\mathbf{A} \times (\mathbf{B} \times \mathbf{C}) = \mathbf{B}(\mathbf{A} \cdot \mathbf{C}) - \mathbf{C}(\mathbf{A} \cdot \mathbf{B}) \quad (2.15)$$

to multiply out the vectors in Eq. (2.12) and get the angular momentum vector solely in terms of scalar components written in body frame as

$$\mathbf{H} = \int_{body} \left\{ \begin{array}{l} \mathbf{b}_1(\omega_1(y^2 + x^2) - \omega_2xy - \omega_3zx) \\ \mathbf{b}_2(-\omega_1xy + \omega_2(x^2 + z^2) - \omega_3zy) \\ \mathbf{b}_3(-\omega_1xz - \omega_2yz + \omega_3(x^2 + y^2)) \end{array} \right\} dm. \quad (2.16)$$

By factoring out the ω_i components in Eq. (2.16), since they do not have any bearing on the integral about the body, Eq. (2.16) can be rewritten as

$$\mathbf{H} = \int_{body} \begin{Bmatrix} y^2 + z^2 & -yx & -zx \\ -xy & x^2 + z^2 & -zy \\ -xz & -yz & x^2 + y^2 \end{Bmatrix} dm \begin{Bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{Bmatrix}. \quad (2.17)$$

Distributing the integral into the 3-by-3 matrix, forms what is commonly referred to as the *mass moment-of-inertia* or *moment-of-inertia* (MOI) matrix which can be expressed as

$$I = \begin{Bmatrix} \int_b (y^2 + z^2) dm & -\int_b yx dm & -\int_b zx dm \\ -\int_b xy dm & \int_b (x^2 + z^2) dm & -\int_b zy dm \\ -\int_b xz dm & -\int_b yz dm & \int_b (x^2 + y^2) dm \end{Bmatrix}. \quad (2.18)$$

The MOI matrix accounts for the mass distribution about the rigid body and is often described as a mass property of the rigid body. Formal names are also given to the diagonal and off-diagonal components of this MOI matrix. The diagonal components of the MOI matrix are called the MOI of the rigid body, while the off-diagonal components are called the *products of inertia* (POI). Methods for determining the mass properties for rigid bodies and their impacts on spacecraft attitude dynamics will be discussed further in Section 2.3.1.

The angular momentum H from Eq. (2.12) can now expressed as

$$\mathbf{H} = I\omega^{bi} \quad (2.19)$$

which is written solely in terms of the MOI matrix I and the angular velocity vector ω^{bi} .

2.1.3 Euler Rotational Equations. The *principle of conservation of angular momentum* requires that the amount of applied angular moment, or torque, applied to a system must be equal to the time rate of change in angular momentum [52]. Therefore,

the resultant moment applied to the body can be written as

$$\mathbf{M} = \frac{d}{dt}\mathbf{H} = \dot{\mathbf{H}} = I\dot{\omega}^{bi} \quad (2.20)$$

where \mathbf{M} is the vector form of the applied torque and $\dot{\mathbf{H}}$ is the time rate of change in angular momentum with respect to the inertial frame, otherwise referred to as the inertial frame derivative of angular momentum.

The inertial frame derivative of $\dot{\mathbf{H}}$ expressed in body frame components is

$$\mathbf{M} = I\dot{\omega}^{bi} + \omega^{bi} \times I\omega^{bi} \quad (2.21)$$

where $\dot{\omega}^{bi}$ is the time rate of change of angular velocity with respect to the body frame. By assuming that body-frame of the rigid body is aligned with its principles axes, then off-diagonal terms of the MOI matrix, also referred to as the POI, are zero such that the MOI matrix is

$$I = \begin{Bmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{Bmatrix}. \quad (2.22)$$

In general, this can be a good assumption for spacecraft dynamics, because most spacecraft are designed with small POI. The effects of POI on spacecraft dynamics is further discussed in Section 2.3.1.

From Eq. (2.22), Eq. (2.21) can be rewritten in vector component form as

$$\mathbf{M} = \begin{Bmatrix} A\dot{\omega}_1 + (C - B)\omega_2\omega_3 \\ B\dot{\omega}_2 + (A - C)\omega_1\omega_3 \\ C\dot{\omega}_3 + (B - A)\omega_1\omega_2 \end{Bmatrix} \quad (2.23)$$

which is often referred to as Euler's rotational equations or Euler's equations, since they were originally developed by mathematician Leonard Euler. Euler's rotational equations are the idealized equations of motion (EOM) for a spacecraft's attitude because the POI were assumed to be zero.

To describe the time rate of change of the spacecraft's angular velocity, Eq. (2.23) can be rewritten as

$$\dot{\omega}_1 = \frac{M_1}{A} + \frac{B - C}{A}\omega_2\omega_3, \quad (2.24)$$

$$\dot{\omega}_2 = \frac{M_2}{B} + \frac{C - A}{B}\omega_1\omega_3, \quad (2.25)$$

and

$$\dot{\omega}_3 = \frac{M_3}{C} + \frac{A - B}{C}\omega_1\omega_2 \quad (2.26)$$

where M_1 , M_2 , and M_3 are the scalar components of the applied torque written in the body frame [54].

2.1.4 Rotational Kinematics. Equations (2.24), (2.25), and (2.26) describe the time rate of change of the spacecraft's angular velocity in the body frame coordinate system. A spacecraft's attitude can be described in the Earth-Centered Inertial (ECI) coordinate system, a non-rotating coordinate frame fixed to the center of the Earth. In order to describe the time rate of change of a spacecraft's attitude, a relationship must be formed to describe the rotational position of the body frame in terms of the ECI coordinate frame. This is accomplished using a set of three successive rotations of the ECI coordinate system, called Euler Angles [30]. The aerospace industry often uses a standard set of Euler Angles to describe the heading of both aircraft and spacecraft as shown in Figure 2.3. This standard set of Euler Angles is often referred to as the Roll-Pitch-Yaw angles of the aircraft or spacecraft. The Roll-Pitch-Yaw Euler Angle rotations can be written in matrix form as

$$\mathbf{C}_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix}, \quad (2.27)$$

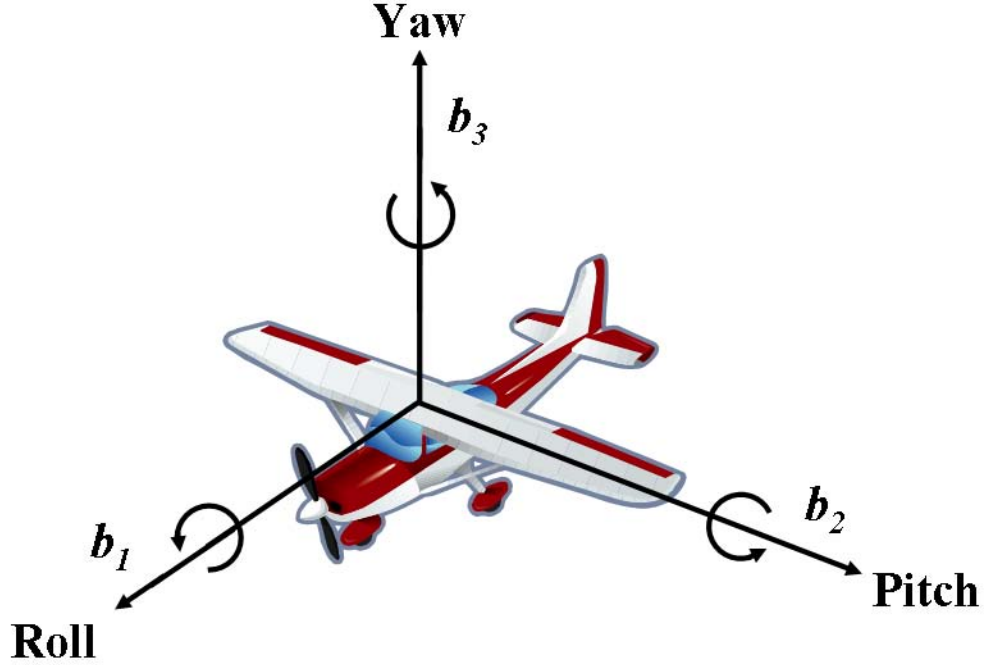


Figure 2.3: Standard Aerospace 1-2-3 Euler Angle Set

$$\mathbf{C}_2(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}, \quad (2.28)$$

and

$$\mathbf{C}_3(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.29)$$

where ϕ is the rotation about the 1-axis, θ is the rotation about the 2-axis, and ψ is the rotation about the 3-axis.

There are four different coordinate frames represented in Figure 2.4. The ECI coordinate frame $(\hat{i}_1, \hat{i}_2, \hat{i}_3)$, the body coordinate frame $(\hat{b}_1, \hat{b}_2, \hat{b}_3)$, and two intermediate coordinate frames $(\hat{a}'_1, \hat{a}'_2, \hat{a}'_3)$ and $(\hat{a}''_1, \hat{a}''_2, \hat{a}''_3)$. The $(\hat{a}'_1, \hat{a}'_2, \hat{a}'_3)$ coordinate frame is formed by the ϕ the rotation about the \hat{i}_1 axis. The $(\hat{a}''_1, \hat{a}''_2, \hat{a}''_3)$ coordinate frame is formed by the θ rotation about the \hat{a}'_2 axis. The $(\hat{b}_1, \hat{b}_2, \hat{b}_3)$ is formed by the ψ rotation about the

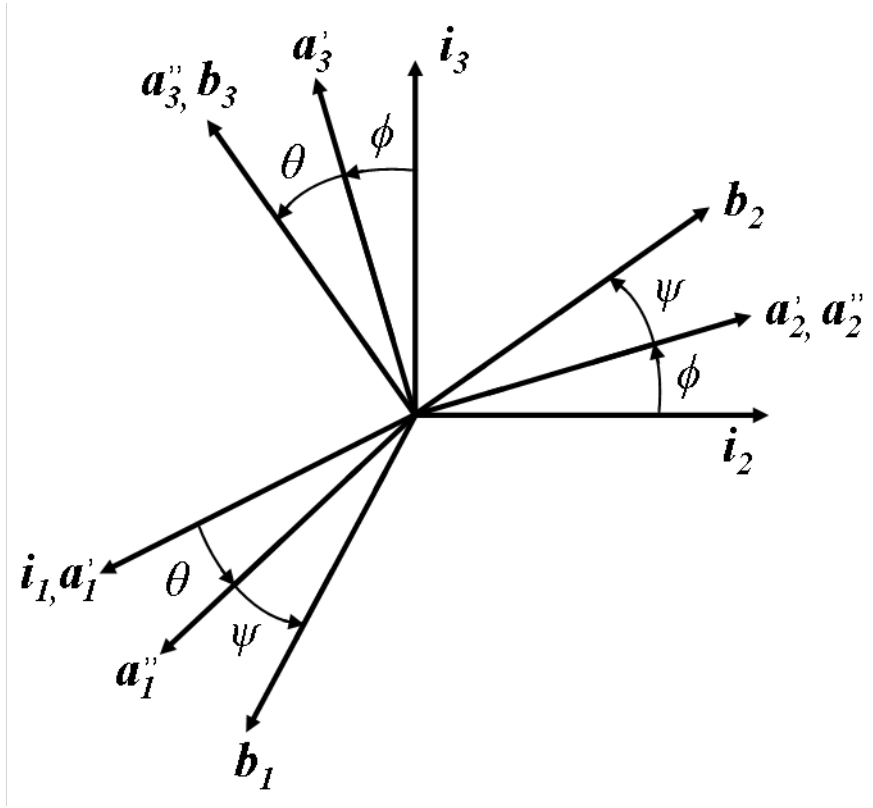


Figure 2.4: 1-2-3 Euler Angle Rotation from Inertial Frame to Body Frame

\hat{a}_3'' axis. Therefore, the total rotation from the inertial frame to the body frame is simple a combination of Eq.'s (2.27), (2.28), and (2.29) written as

$$\mathbf{C}^{B/I} = \mathbf{C}_3(\psi)\mathbf{C}_2(\theta)\mathbf{C}_1(\phi). \quad (2.30)$$

Notice that the order the matrix multiplication in Eq. (2.30) is the reverse of how the Euler Angle set is described. This is due to the intermediate coordinate frames that are formed after successive rotations.

The body frame angular velocity can now be written as

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \mathbf{C}_3(\psi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{C}_3(\psi)\mathbf{C}_2(\theta) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \quad (2.31)$$

where $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ is the time rate of change of the Roll-Pitch-Yaw Euler Angles. Eq. (2.31) can also be rewritten in the body frame as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos \theta} \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \theta \cos \theta & \cos \theta \cos \psi & 0 \\ -\sin \theta \cos \psi & -\sin \theta \sin \psi & \cos \theta \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \quad (2.32)$$

2.1.5 Quaternion Parameters. An inconvenience of using Euler Angle rotation matrices, as defined in Section 2.1.4, is that they have an inherent geometric singularity. It is clear from Eq. (2.32), that for the Roll-Pitch-Yaw set of Euler Angles, this singularity occurs when rotation θ about the \hat{i}_2 axis approaches $n(\pi/2)$ radian, where n is an odd integer. This singularity does not physically exist, and only exists as a result of the definition of Euler Angles. However, when using Euler Angles to define the rotational relationship between the ECI and body frame, the singularity must be avoided. SimSAT II cannot reach the angle of singularity due to the physical constraints imposed by its air bearing, however, satellites on orbit do not have these physical constraints. In order to maintain realism between the SimSAT II simulated satellite system and an actual satellite, methods for avoiding these singularities are addressed.

To avoid the Euler Angle singularity quaternion parameters are used, providing a convenient singularity-free alternative to define a spacecraft's attitude. The drawback to using quaternion parameters is that their physical interpretation is less readily apparent as compared to Euler Angles [52].

Quaternion parameterization begins by first defining the rotation from of a spacecraft from one reference frame to another in terms of a rotation matrix such as the one used in Eq. (2.30). The Euler axis of rotation is common to both frames of reference, before and after the rotation, and is expressed in scalar quantities common to both frames such as

$$\mathbf{e} = e_1 \hat{i}_1 + e_2 \hat{i}_2 + e_3 \hat{i}_3 \quad (2.33)$$

and

$$\mathbf{e} = e_1 \hat{b}_1 + e_2 \hat{b}_2 + e_3 \hat{b}_3 \quad (2.34)$$

where \mathbf{e} is the Euler axis of rotation. The L norm of \mathbf{e} is also subject to the constraint,

$$\|\mathbf{e}\|_2 = 1. \quad (2.35)$$

The calculation of the scalar value for e_i is simplified considerably when the desired rotation is about an axis along the direction vector shared by the two reference frames, such as in Figure 2.5. This is because the scalar value of e_i about the shared axis is one, while the other two values of e_i are zero. The rotation angle for the corresponding Euler

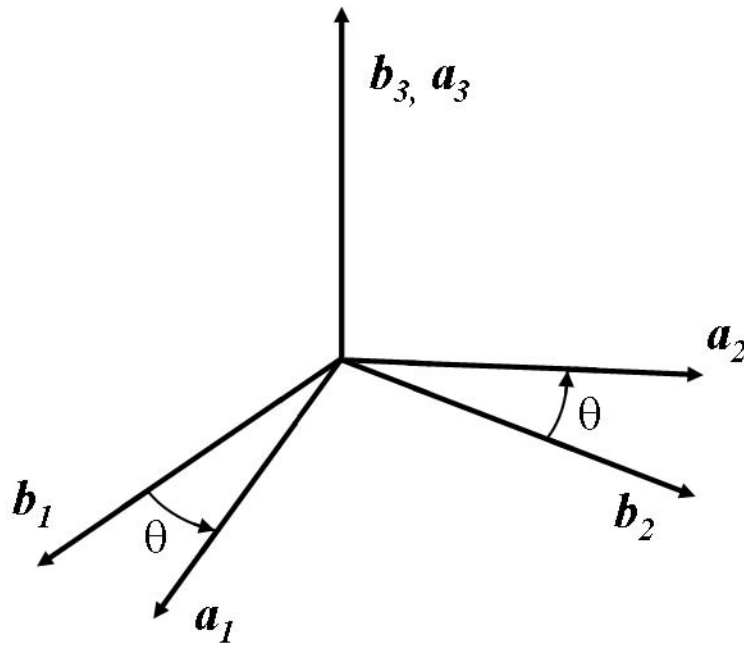


Figure 2.5: Rotation about a Direction Vector Shared by Reference Frames

axis of rotation can be found from Eq. (2.36),

$$\cos \theta = \frac{1}{2}(C_{11} + C_{22} + C_{33}) \quad (2.36)$$

where θ is the rotation angle. With values for both θ and e_i , the quaternion parameters are calculated from

$$q_1 = e_1 \sin\left(\frac{\theta}{2}\right), \quad (2.37)$$

$$q_2 = e_2 \sin\left(\frac{\theta}{2}\right), \quad (2.38)$$

$$q_3 = e_3 \sin\left(\frac{\theta}{2}\right), \quad (2.39)$$

$$\text{and } q_4 = \cos\left(\frac{\theta}{2}\right). \quad (2.40)$$

Due to the constraint on the Euler axis of rotation \mathbf{e} in Eq. (2.35), the L norm constraint of the set of quaternion parameters found in Eq. (2.37) through Eq. (2.40) also implies

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1. \quad (2.41)$$

Going back to the example, where \mathbf{e} corresponds to a direction vector shared by the two reference frames, the calculation of the quaternion parameters is simplified because two of the four quaternions must be zero. Also similar to Euler axis of rotation \mathbf{e} , the first three quaternion parameters are sometimes referred to in vector format as shown in

$$\mathbf{q} = (q_1, q_2, q_3) \quad (2.42)$$

while q_4 is always referred to as a separate scalar.

In addition to being referred to in terms of Euler angles and axes, direction cosine matrices can also be referred to in terms of quaternions as

$$\mathbf{C}^{ba} = \mathbf{C}(\mathbf{e}, \theta) \quad (2.43)$$

or

$$\mathbf{C}^{ba} = \mathbf{C}(\mathbf{q}, q_4). \quad (2.44)$$

Successive direction cosine matrix rotations expressed in terms of quaternions can also be multiplied similar to to direction cosine matrices expressed in terms of eigenaxes and

eigenangles, shown as

$$\mathbf{C}^{ba} = \mathbf{C}(\mathbf{e}, \theta)^{ba''} \mathbf{C}(\mathbf{e}, \theta)^{a''a'} \mathbf{C}(\mathbf{e}, \theta)^{a'i} \quad (2.45)$$

or

$$\mathbf{C}^{ba} = \mathbf{C}(\mathbf{q}, q_4)^{ba''} \mathbf{C}(\mathbf{q}, q_4)^{a''a'} \mathbf{C}(\mathbf{q}, q_4)^{a'i}. \quad (2.46)$$

To directly calculate the total quaternion rotation formed by two successive direction cosine matrices, one can use

$$\mathbf{q} = q_4'' \mathbf{q}' + q_4' \mathbf{q}'' + \mathbf{q}' \times \mathbf{q}'' \quad (2.47)$$

and

$$q_4 = q_4' q_4'' - (\mathbf{q}')^T \mathbf{q}'' \quad (2.48)$$

where q_i' represents the quaternions for the first rotation, q_i'' represent the quaternions for the second rotation, and q_i represent the total quaternion rotation. Eq. (2.47) and Eq. (2.48) are equivalent to

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} q_4'' & q_3'' & -q_2'' & q_1'' \\ -q_3'' & q_4'' & q_1'' & q_2'' \\ q_2'' & -q_1'' & q_4'' & q_3'' \\ -q_1'' & -q_2'' & -q_3'' & q_4'' \end{bmatrix} \begin{bmatrix} q_1' \\ q_2' \\ q_3' \\ q_4' \end{bmatrix} \quad (2.49)$$

where the orthogonal 4-by-4 matrix is sometimes referred to as the quaternion matrix. The quaternions formed from the Roll-Pitch-Yaw Euler Angle set described in in Section 2.1.4 are written as

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} c_\psi c_\theta s_\phi + s_\psi s_\theta c_\phi \\ c_\psi s_\theta c_\phi - s_\psi c_\theta s_\phi \\ c_\psi s_\theta s_\phi + s_\psi c_\theta c_\phi \\ c_\psi c_\theta c_\phi - s_\psi s_\theta s_\phi \end{bmatrix} \quad (2.50)$$

where $s_{(-)}$ and $c_{(-)}$ represent $\sin(-/2)$ and $\cos(-/2)$, respectively.

When using quaternions in control systems, it is often necessary to calculate the quaternion error \mathbf{q}_e between the quaternions parameters for the current orientation \mathbf{q}_c and the quaternions parameters for the desired orientation \mathbf{q}_d . Since quaternions are four dimensional parameters, the quaternion error vector cannot be calculated simply by subtracting \mathbf{q}_c from \mathbf{q}_d . Instead, with a slight modification, Eq. (2.49) can be written as

$$\begin{bmatrix} q_{e,1} \\ q_{e,2} \\ q_{e,3} \\ q_{e,4} \end{bmatrix} = \begin{bmatrix} q''_{d,4} & q''_{d,3} & -q''_{d,2} & q''_{d,1} \\ -q''_{d,3} & q''_{d,4} & q''_{d,1} & q''_{d,2} \\ q''_{d,2} & -q''_{d,1} & q''_{d,4} & q''_{d,3} \\ -q''_{d,1} & -q''_{d,2} & -q''_{d,3} & q''_{d,4} \end{bmatrix} \begin{bmatrix} -q_{c,1} \\ -q_{c,2} \\ -q_{c,3} \\ q_{c,4} \end{bmatrix}. \quad (2.51)$$

When the current orientation \mathbf{q}_c is equal to the desired orientation \mathbf{q}_d , \mathbf{q}_e takes the form $[0, 0, 0, 1]$ [50].

The quaternion parameters are related to the angular velocity and current quaternion states by

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & \omega_3 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}. \quad (2.52)$$

This relationship is used in place of Euler's equations for this research, because quaternion parameters are used instead of Euler Angles [52].

2.2 *Satellite Simulators*

Satellite simulators have been used since the 1960's to perform terrestrial-based testing of spacecraft attitude control systems prior to launch [46]. Spacecraft operate in a low gravity, vacuum environment with normally uncoupled rotational and translational DOFs. Satellite simulators are characterized by the approach they take to simulate torque-free on-orbit conditions.

2.2.1 Historical Simulated Satellites. The most widely used method to simulate on-orbit conditions for spacecraft attitude control testing is through use of an air bearing. Air bearings function by separating two smoothed surfaces by a cushion of pressurized air, creating a near frictionless boundary. They are typically divided into two categories: planar and rotational. A planar air bearing would resemble an air-hockey table, allowing for two translational DOF motion and one rotational DOF. While rotational air bearings resemble a ball-in-socket joint, allowing for three DOF rotational motion but has no translational DOF.



Figure 2.6: AEROTECH AHL9000 Series Planar Air Bearing [3]

Planar air bearings, such as the one shown in Figure 2.6, may resemble an air-hockey table, however the table surface of a planar air bearing usually does not supply the pressurized air. Instead, most test bodies carry their own supply of air to create a near frictionless condition on the planar air bearing's smoothed table surface. The planar motion enabled by these air bearing is well suited for simulating docking and rendezvous maneuvers [46]. As far back as 1967, records indicated that NASA has used planar air bearings for both manned and unmanned vehicle testing [24]. However, more recent use of planar air bearings have centered around control testing of robotic arms and other extendible spacecraft appendages [45] [35] [38] [37]. Other recent uses of planar air bearings has included the testing of control algorithms for a variety of microsattellites [49] [15] [17] [29].

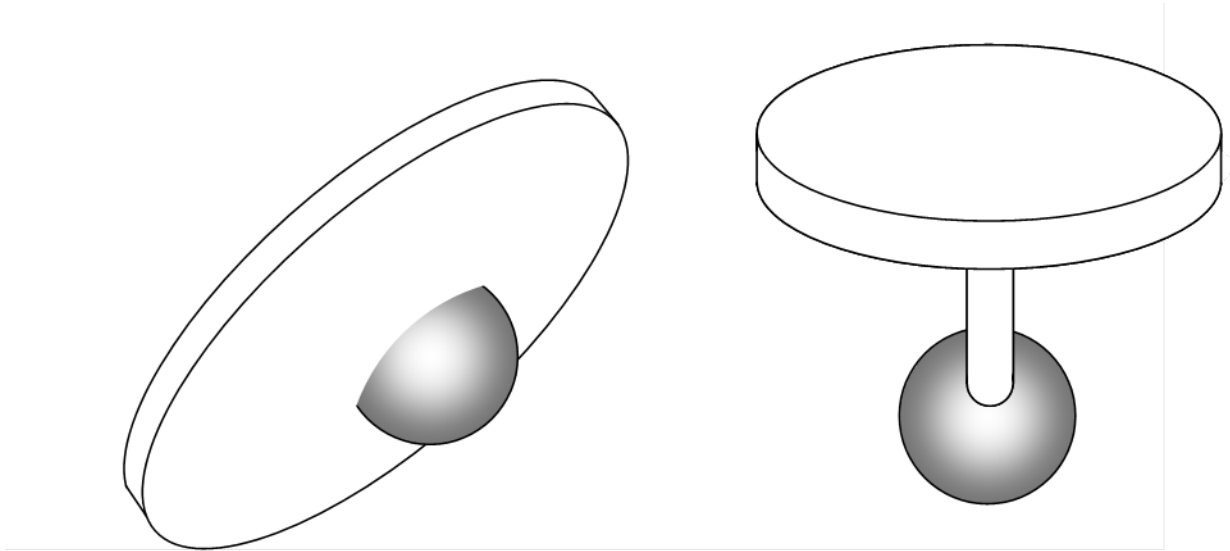


Figure 2.7: “Tabletop” and “Umbrella” Spherical Air Bearing Configurations

As mentioned before, rotational air bearings, sometimes called spherical air bearings, are the most widely used system for testing spacecraft attitude control. By allowing three rotational DOF a simulated satellite testbed can effectively mimic a torque-free environment. The torque-free environment of a spherical air bearing is dependent on the alignment of the center of mass of a simulated satellite with the spherical air bearing’s center of rotation. The structural configuration of a simulated satellite’s hardware relative to its air bearing’s spherical ball bearing determines the freedom of spin about any particular axis [46]. Currently and as far back as early 1960’s, these testbeds have typically been a “tabletop” or an “umbrella” configuration, as shown in Figure 2.7. This design enables uninhibited rotation about its Yaw axis, but has limited rotation about the other two axes [27] [20] [34] [18]. More recently, several spacecraft control testbeds have moved to a “dumbbell” configuration as shown in Figure 2.8, to enable full freedom of spin about the Yaw and Roll axis with limited rotation only about one axis [47] [14] [16].

2.2.2 AFIT SimSATs. AFIT has designed, built, and tested two simulated satellites since 1999. AFIT’s first simulated satellite, SimSAT I, shown in Figure 2.9, was designed and built in 1999 using the “dumbbell” configuration [16]. Between 1999 and 2007, a wide variety of dynamics and control research efforts were conducted using

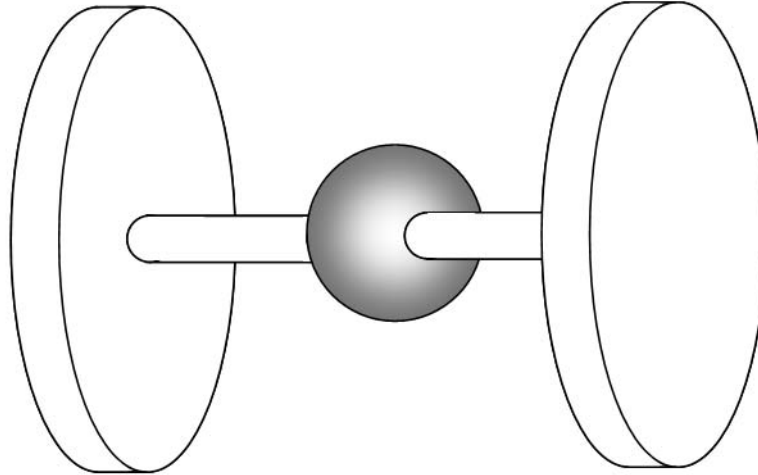


Figure 2.8: “Dumbbell” Spherical Air Bearing Configuration

SimSAT I, including [19] [25] [48] [26] [30]. However, SimSAT I had limitations as noted by Rohe, Roach and Welty.

...SimSAT did have limitations. Due to its large mass and inertia, it was not able to conduct rapid slew maneuvers or to spin up to the required angular velocity necessary to achieve spin stabilization. Also, its momentum wheels quickly saturated while attempting to move the massive simulator. Further, the dumbbell experienced structural deflections that cause its center of mass to move when the simulator rotated about its roll axis. The aging hardware on the SimSAT was also growing out of date and was in need of replacement. As a result of these limitation, AFIT faculty requested that an improved satellite simulator be designed.

Because this research is based around the use of AFIT’s second satellite simulator, SimSAT II, the design details are left to the test description part of this document, Section 3.1.

2.3 System Characterization

2.3.1 Mass Properties Determination. Accurate knowledge of spacecraft mass properties, or the lack thereof, can have a significant impact on nearly all phases of a spacecraft’s lifecycle, from initial launch to routine attitude control and orbit transfer



Figure 2.9: AFIT's SimSAT I

[11]. The mass properties of a spacecraft are its mass, the location of its CG, and its MOI matrix.

The mass of a spacecraft is a primary consideration when selecting a launch system. Current spacecraft systems engineering practices suggest selecting a desired class of launch vehicles instead of selecting a particular launch vehicle in order to increase the probability of meeting desired schedule and cost [51]. To manufacture a spacecraft compatible with multiple launch vehicles, stringent mass property requirements must be kept. SimSAT II does not have any launch systems based mass requirements, however it does have mass requirements based on the maximum loading capabilities of the spherical air bearing. The estimated total mass of SimSAT II is 40% of the maximum loading capability of the spherical air bearing currently in use. Therefore, it is not necessary at present to undergo the rigorous methods for mass calculation suggested by Boynton [10] and [12]. Before testing, SimSAT II's mass did fluctuate slightly from adding and removing counterweights for balancing purposes. All testing was conducted using the same fixed set of counterweights, however small position adjustable masses were still manipulated for fine balancing purposes.

A satellite floating on a spherical air bearing must have its CG at the center of rotation (CR) of the air bearing in order to accurately simulate a torque-free environment [9]. If the satellite simulator's CG is located above the CR, the spacecraft is in an unstable configuration that will tilt due to gravity causing a torque about the CR. This unstable configuration is very similar to an inverted pendulum tilting away from its unstable equilibrium point as seen in Figure 2.10. When the spacecraft's CG is located below the center of rotation, the spacecraft will tend to oscillate and eventually come to rest with its center of mass at its lowest equilibrium point, similar to a pendulum oscillating about its stable equilibrium or a spacecraft oscillating due to a gravity gradient torque [43]. Boynton suggests counterbalancing the satellite or splitting the satellite into two halves to ensure CG is located at the CR of the air bearing [9]. Ross on the other hand, suggests actively compensating for the offset between the spacecraft CG and the air bearing center of rotation in the spacecraft's EOM [43].

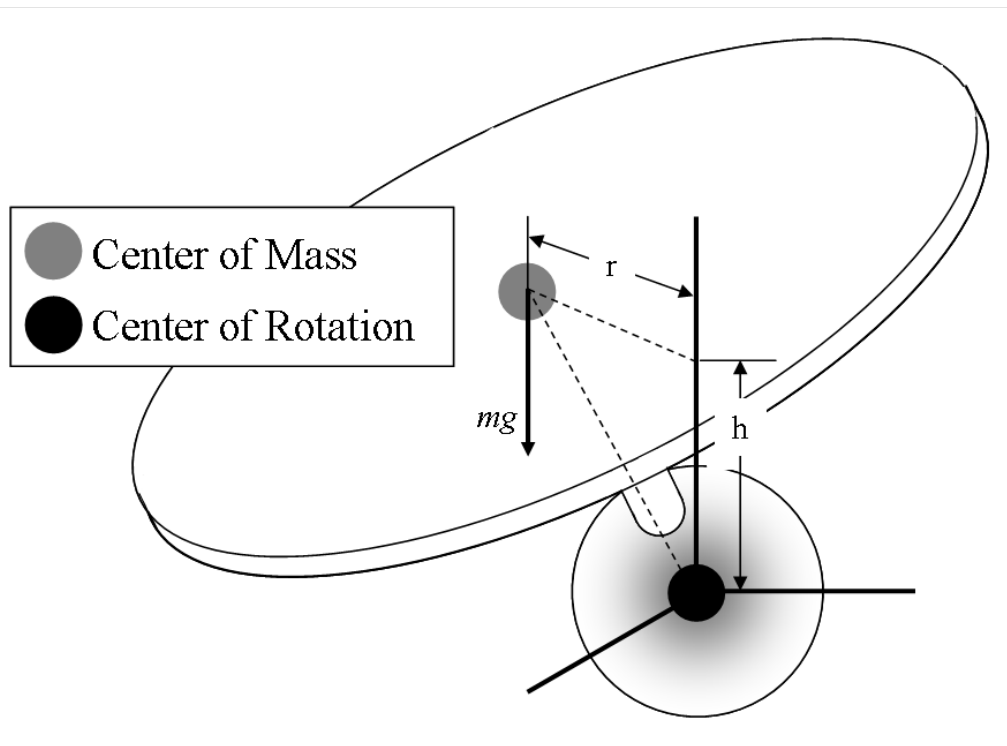


Figure 2.10: Misalignment of CG and CR Causing Torque on a Rigid Body

For attitude control problems, the mass of a spacecraft is not explicitly expressed in its EOMs. However, the distribution of mass does have a significant impact on a

spacecraft's dynamics. As previously described in Section 2.1.2, a volume integral of a rigid body can be thought of as an integral taken for a rigid body partitioned into a large number of small volumes of mass. The mass of each of the small volumes implicitly impacts the magnitude of its MOI and POI as calculated in Eq. (2.18). Therefore, mass implicitly impacts a spacecraft dynamics through the magnitude of its MOI and POI.

The magnitude of the POI for a spacecraft can have a significant impact on its ACS. Spacecraft with relatively large POI terms with respect to the diagonal terms in the MOI matrix cannot use the assumption in Eq. (2.22) that the POI are zero or can be ignored. By including non-zero POI terms, the complexity of the ACS increases because it must now account for all the cross terms that could be ignored by assuming the POI are zero. Without correct compensation in its ACS, a spacecraft with large POI in its MOI matrix will experience a non-zero ω_i about its un-torqued axes due to these cross terms. In other words, if one uses attitude thrusters to spin a spacecraft with large POI about one of its body frame axis, then this spacecraft will also experience rotations about its other axes as well. This is especially problematic for spin-stabilized spacecraft that are continuously spinning about one of its body frame axes. SimSAT II's POI are assumed to be zero since they are small enough to cause torques less than 10^{-6} N-m.

2.3.2 Thruster Characterization. SimSAT II has thruster simulators, electric motors and propellers, as its only attitude control actuator. In order to discuss SimSAT II's thruster simulators, fundamental aerodynamic theory must first be discussed. The force \mathbf{F} generated by the spinning of a propeller can be calculated from

$$\mathbf{F} = A\Delta\mathbf{p} \tag{2.53}$$

where $\Delta\mathbf{p}$ is the change in pressure over circular area A spanned by a propeller as it is spinning [13]. From Bernoulli's equation

$$p + \frac{1}{2}\rho V^2 + \rho gh = constant \tag{2.54}$$

a constant relationship is defined between pressure p , the velocity of an object V traveling through a fluid of constant density ρ , and the pressure created in a fluid from its density, the acceleration due to gravity g , and altitude h .

To calculate a change in pressure between two points, assuming constant density as shown in Fig. 2.11, then Eq. (2.54) at the initial point must be subtracted from Eq. (2.54) at the final point. This results in the dynamics pressure equation

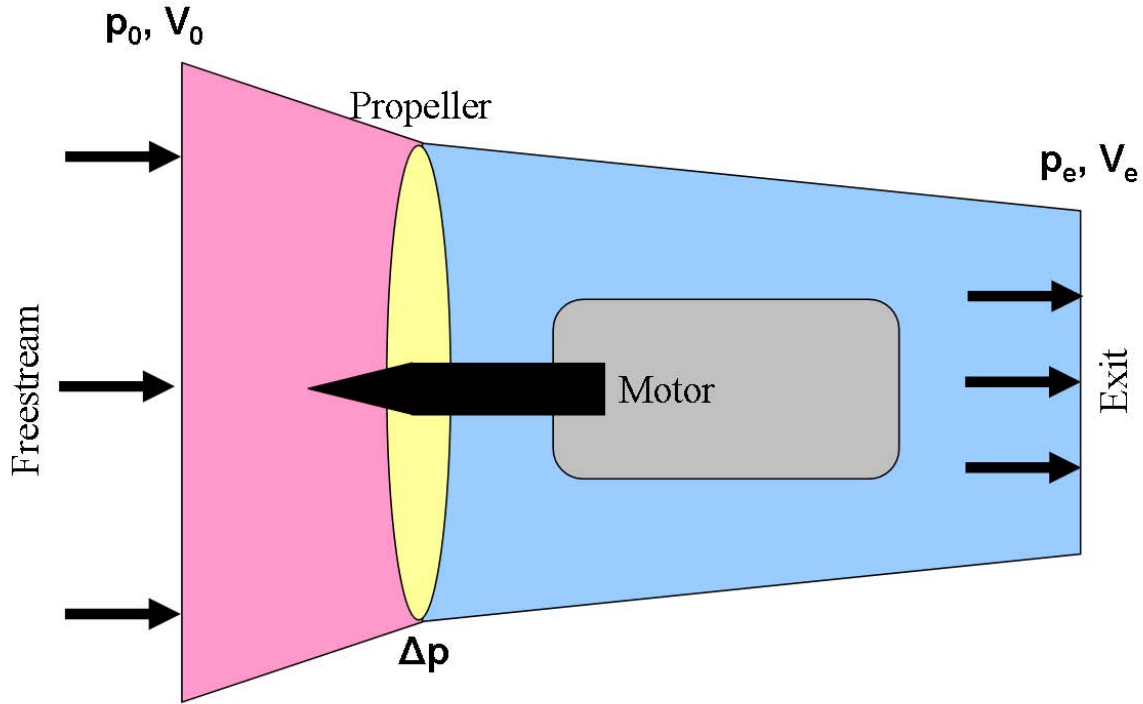


Figure 2.11: Idealized Propeller Thrust Production

$$\Delta p = \frac{1}{2}\rho (V_e^2 - V_0^2) \quad (2.55)$$

where Δp is the change in pressure between the two points, V_0 is the velocity of the fluid at the initial point, and V_e is the velocity of the fluid at the final point. By substituting Δp in Eq. (2.55) into Eq. (2.53), an idealized equation is formed for the force from or thrust created by a propeller in terms of the velocity of the airflow in front of and behind the propeller.

$$\mathbf{F} = \frac{1}{2}\rho A(\mathbf{V}_e^2 - \mathbf{V}_0^2). \quad (2.56)$$

Equation (2.56) provides an ideal calculation for thrust, but it does not account for losses such as air drag, changes in inflow velocity, temperature gradients, fluid compression, etc [13]. To account for these losses when determining the theoretical thrust of a propeller, empirically derived efficiency and coefficients factors can be used to achieve more accurate results [28]. However, there is little to no performance characteristics research data for the small diameter propellers being used on SimSAT II [21]. This is largely due to the fundamental difference in fluid dynamic behavior involved with small diameter propellers and those found on man-rated or large aircraft. This difference in fluid dynamic behavior is characterized by the non-dimensional coefficient called the Reynolds number Re .

Small diameter propellers typically operate under small Reynolds number flow conditions ($Re < 500,000$). When the Reynolds number is much less than 500,000, the viscous forces are much greater than the inertial forces due to the fluid, resulting in more laminar flow around the propeller. As a result, small propellers experience unsteady laminar flow separation, making the thrust produced by small propeller less predictable than larger propellers experiencing flow with a high Reynolds number [36].

The Reynolds number for SimSAT II's propellers can quickly be verified by

$$Re = \frac{\rho_{\infty} V_{\infty} x}{\mu_{\infty}} \quad (2.57)$$

using sea level conditions for the freestream density ρ_{∞} , 1.23 kg/m³, and freestream viscosity μ_{∞} , 1.79×10^{-5} kg/m · s, using the maximum linear velocity of the propeller for freestream velocity V_{∞} , 3.14 m/s, and using the full propeller chord length as the critical point of the propeller x , 0.01 m [4]. From these assumptions, the Reynolds number for the propellers on SimSAT II is approximately 2158 and will exhibit performance characteristics similar to other propellers with Reynolds numbers much less than 500,000. Since there is a lack of detailed performance data for these types of propellers, propeller characterization testing is the only reliable way to predict the forces generated by the propellers. Therefore, one of the initial steps taken in this research is detailed fan characterization testing.

2.4 *Optimal Control Theory*

Optimization, per *Merriam-Webster's Collegiate Dictionary* [1], is

op·ti·mi·za·tion *n* (1857) : an act, process, or methodology of making something (as a design, system, or decision) as fully perfect, functional, or effective as possible; *specif*: the mathematical procedures (as finding the maximum of a function) involved in this.

In the context of engineering and mathematics, the definition of optimization can typically be reduced to finding the desired extrema of a function. Therefore, optimal design is defined as the general concept of using mathematical theorem and methods to find the extrema of a function that represents the optimality of a system's design.

Optimal design problems use a standard mathematical notation, regardless of engineering or mathematical discipline [5]. Optimal design problems are typically separated into three major components: design variables, constraint functions, and the cost function. Design variables \mathbf{x} are independent parameters that represent the system being optimized and can be manipulated to achieve optimality. Constraints functions $h(\mathbf{x})$ and $g(\mathbf{x})$ are equality or inequality functions, respectively, that must be satisfied in order for an optimal solution to be feasible and constrain the value of one or more design variable. The cost function $f(\mathbf{x})$, sometimes referred to as the performance index or objective function J , is a function of one or more of the design variables whose extrema represents the goal of optimum design. For more detailed discussion on methods for transcribing real world systems into formulated optimum design problems, the reader is referred to [5] and [6].

After correctly formulating an optimal design problem, it must then be classified in order to determine the solution method. While generality does exist with respect to the mathematical theorems and methods used for solving the different classes of optimal design problems, a specialized set of theorems and methods is typically used for each classification. The only classification of optimal design problems that will be discussed in great detail is the fixed initial time, free final time nonlinear optimal control problem. Other classifications of optimal design problems will be discussed in brevity and only as required.

Optimal design problems are typically separated into two classes: linear and non-linear. If both the constraint functions and cost function contain all linear terms, then the problem is classified as a linear optimal design problem. Linear optimal design problems can be solved using linear programming (LP) techniques. However, most aerospace optimal design problems are not linear. If any of the constraint functions or the cost function contain nonlinear terms, then the entire optimal design problem is considered a nonlinear programming (NLP) problem. There are two methods for solving NLP problems: indirect and direct methods. Indirect methods for solving NLP problem involve minimization techniques involving calculus of variation methods to obtain an analytical solution. On the other hand, direct methods for solving NLP problems involve numerical search techniques to iteratively obtain an optimal solution. The preferred method used to solve a NLP problem is dependent on the problem being solved. Since it is not always feasible to find an analytical solution, the use of indirect methods is generally restricted to simple problems, with few states and constraints. Direct methods are used for most aerospace problems because of their highly coupled and nonlinear nature. This research uses a hybrid approach that formulates the problem using indirect methods and then solves the problem using direct methods [31].

An optimal design problem is considered an optimal control problem when its constraint functions $h(\mathbf{x})$ are differential equations describing the system's dynamics. These constraints typically contain undifferentiated design variables \mathbf{u} , referred to as controls, which manipulate the system's dynamics. Due to differential constraints, the performance index J of an optimal control problem is described by a cost functional because the solution to the performance index is now described by a function of functions instead of a vector. The ultimate goal of optimal control is to find a function of the control history that minimizes the problem's performance index functional while adhering to all of the system's dynamical constraints [31]. For this research, the system dynamics of SimSAT II's attitude in quaternion parameters is described by Eq. (2.52) and the system dynamics for SimSAT II's angular rates is described by Eq. (2.23). The torque generated by SimSAT II's thruster simulators is described by the control variable \mathbf{u} .

2.4.1 Fixed Initial Time, Free Final Time Optimal Control. This section outlines the particular methods for formulating and solving a generalized fixed initial time, free final time optimal control problem. A generalized fixed initial time, free final time optimal control problem can be solved using either the indirect or direct methods approach. This problem will be solved using indirect methods to find an analytical solution. While deriving an indirect solution, it will be identified where it is mathematically infeasible to use the indirect methods and preferable to use direct methods. The direct methods approach used in this research will be described in the next section.

The performance index functional for a generalized fixed initial time, free final time optimal control is written as

$$J = \phi(t_f, \mathbf{x}_f) + \int_{t_0}^{t_f} L(t, \mathbf{x}, \mathbf{u})dt, \quad (2.58)$$

where ϕ is a function of the final time and state conditions and L is an integral cost function [31]. The generalized system dynamics for this problem are described by the differential constraints

$$\dot{\mathbf{x}} = f(t, \mathbf{x}, \mathbf{u}). \quad (2.59)$$

This problem is subject to the initial condition constraints written as

$$t_0 = t_0^*, \mathbf{x}_0 = \mathbf{x}_0^*, \quad (2.60)$$

where t_0 and \mathbf{x}_0 are the initial time and states, respectively, and the problem must contain initial condition constraints t_0^* and \mathbf{x}_0^* for all states. The optimal control problem is also subject to the final condition constraints written as

$$\psi(t_f, \mathbf{x}_f) = \mathbf{x}_f - \mathbf{x}_f^* = 0, \quad (2.61)$$

where ψ must contain at least one state with final condition \mathbf{x}_f^* .

An optimal solution to this problem satisfies the first and second differential conditions for the augmented cost functional

$$J' = G(t_f, \mathbf{x}_f, \nu) + \int_{t_0}^{t_f} [H(t, \mathbf{x}, \mathbf{u}) - \lambda^T \dot{\mathbf{x}}] dt, \quad (2.62)$$

where G is the endpoint function and H is the Hamiltonian [31]. The endpoint functional is written as

$$G = \phi(t_f, \mathbf{x}_f) + \nu^T \psi(t_f, \mathbf{x}_f), \quad (2.63)$$

where ν^T is a vector of constant multipliers for each final condition constraint. The Hamiltonian is written as

$$H = L(t, \mathbf{x}, \mathbf{u}) + \lambda^T f(t, \mathbf{x}, \mathbf{u}), \quad (2.64)$$

where λ^T is a vector of Lagrange multipliers for each differential constraint.

The first differential necessary condition for optimality of the fixed initial time, free final time problem is that the problem must satisfy the Euler-Lagrange (E-L) equations [31]. The E-L equations are formed by three sets of equations: the state, costate, and stationary equations. The state, costate, and stationary equations are derived from the augmented performance index J' and written respectively as

$$\dot{\mathbf{x}} = H_{\lambda}^T = f(t, \mathbf{x}, \mathbf{u}), \quad (2.65)$$

$$\dot{\lambda} = -H_x^T(t, \mathbf{x}, \mathbf{u}, \lambda), \quad (2.66)$$

and

$$0 = H_u^T(t, \mathbf{x}, \mathbf{u}, \lambda), \quad (2.67)$$

where $H_{(-)}$ is the partial differential of H with respect to $(-)$. The E-L equations are subject to the derived boundary conditions in Eq. (2.60) and Eq. (2.61) as well as the

constraints

$$G_{t_f}(t_f, \mathbf{x}_f, \nu) + L(t_f, \mathbf{x}_f, \mathbf{u}_f) + G_{x_f}(t_f, \mathbf{x}_f, \nu)f(t_f, \mathbf{x}_f, \mathbf{u}_f) = 0 \quad (2.68)$$

and

$$\lambda_f = G_{x_f}^T(t_f, \mathbf{x}_f, \nu) \quad (2.69)$$

where $G_{(-)}$ is the partial differential of G with respect to $(-)$. For a problem with n states and m controls, $2n$ differential equations from Eq. (2.65) and Eq. (2.66) and m equations from Eq. (2.67) are solved while satisfying the n initial condition constraints in Eq. (2.60) and n final conditions found by combining Eq. (2.61), Eq. (2.68), and Eq. (2.69) [31]. When a system's dynamical constraints are characterized by nonlinear coupled differential equations, an analytical solution may not exist. For this research, an analytical solution cannot be found using indirect methods because SimSAT II's dynamics are represented by seven coupled, nonlinear differential equations from Eq. (2.52) and Eq. (2.23). To solve optimal control problems for SimSAT II, this research uses DIDO, a pseudospectral-based direct method optimal control solver tool.

Assuming that an analytical solution exists for the system dynamics of a fixed initial time, free final time problem, then the solution must meet the Weierstrass first order necessary condition and the Legendre-Clebsch second order sufficient condition in order to be an optimal solution [31]. The Weierstrass condition written as

$$H(t, \mathbf{x}, \mathbf{u}_*, \lambda) - H(t, \mathbf{x}, \mathbf{u}, \lambda) > 0 \quad (2.70)$$

must be satisfied for all $\mathbf{u}_* \neq \mathbf{u}$, where \mathbf{u} is the optimal control solution and \mathbf{u}_* is all admissible comparison paths. In other words, the Hamiltonian H must be minimized along the optimal control solution. The Legendre-Clebsch condition written as

$$H_{uu}(t, \mathbf{x}, \mathbf{u}, \lambda) \geq 0 \quad (2.71)$$

must also be satisfied at every point along the optimal control solution [31].

2.4.2 *Pseudospectral Methods.* This research uses the DIDO optimal control solver tool to find optimal solutions to attitude maneuvers for SimSAT II. Pseudospectral methods, sometimes referred to as orthogonal collocation methods, is a technique for discretizing the states and controls for an optimal control problem onto a Lagrange interpolating polynomial. There are many distribution methods for the discretized nodal locations on the Lagrange interpolating polynomial including: uniform distribution, Legendre-Gauss points, Legendre-Gauss-Radau points, Legendre-Gauss-Lobatto (LGL) points, Chebyshev-Gauss points, Chebyshev-Gauss-Radau, and Chebyshev-Gauss-Lobatto points [23]. DIDO uses LGL points because it evaluates finite horizon problems, since the number of nodes used to interpolate the Lagrange polynomial is fixed [22]. From this method, DIDO uses the discretized approximation of the states and controls to form a new approximated cost function. The approximated cost function is minimized by DIDO, resulting in a discretized optimal solution. The accuracy of the discretized optimal solution can be increased as number of LGL points used to interpolate the Lagrange polynomial is increased.

Fleming *et al.* used pseudospectral methods to examine the minimum-time, three-independent torque reorientation of an asymmetric rigid body [22]. Their work extended the earlier results of Bilimoria and Wie that an eigenaxis maneuver is not the minimum-time rest-to-rest maneuver for an axis-symmetric spacecraft, by showing that the eigenaxis maneuver is not an optimal solution for an asymmetric rigid body as well [53]. In addition to finding the typical open-loop optimal control, Fleming *et al.* went on to use RTOC to recompute open loop optimal control solutions mid-maneuver similar to a traditional closed-loop controller changing its control magnitude as a result of state measurements. New control solutions were updated into the simulation as soon as they were available. They demonstrated the optimality of each new solution using Bellman's principle of optimality which simply states that the optimal solution from any location on the optimal path is on the optimal path of the original solution. This principle is demonstrated in Figure 2.12. Through the use of RTOC, Fleming *et al.* were able to adapt the optimal control solutions to parametric uncertainty that would have otherwise been impossible using a traditional open-loop optimal control solution.

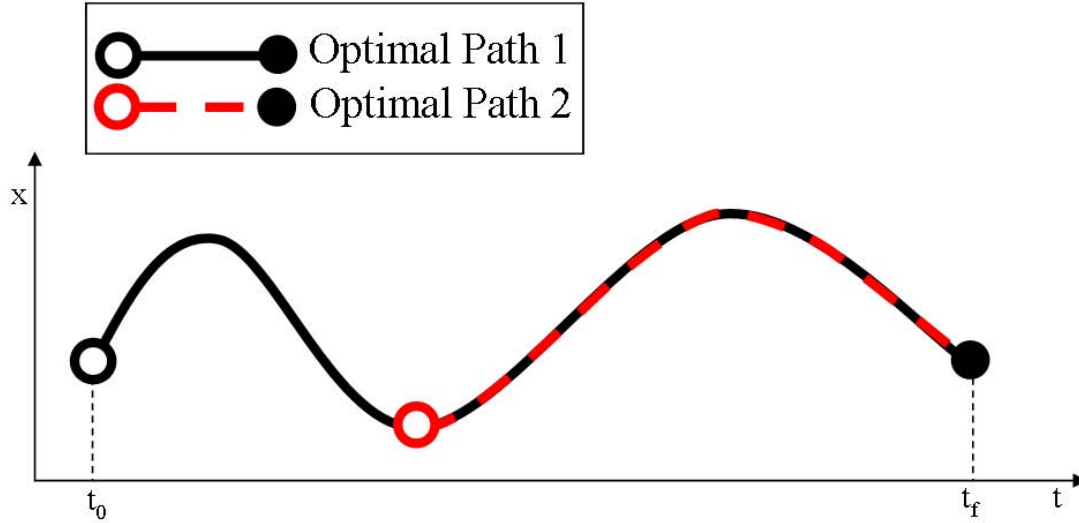


Figure 2.12: Example of Bellman's Principle of Optimality

2.4.3 Real-Time Optimal Control. The foundations for what is now referred to as RTOC, was presented by Ross *et al.* in 2006, based on their results from implementing a RTOC controller on the Naval Postgraduate School's (NPS) satellite simulator, NPSAT1[43]. They outlined a simple sample-and-hold technique for calculating the closed-loop optimal control solution by sampling state and control measurements at fixed intervals, computing the optimal open-loop control solution for each interval, then connecting each solution piecewise to create a closed loop solution. The outer control loop used to continuously recompute optimal control solutions is shown in Figure 2.13. Ross *et al.* also addressed some of the difficulties that arise from creating a piecemeal control trajectory including: control solution discontinuity created between sample intervals, computational delay created while calculating new optimal open-loop solution, and the balance needed between sensor precision and the optimality of updated control solutions.

The success of earlier research has led Ross and others to examine new approaches in trading off computational costs for solution optimality in what they term as the Bellman Pseudospectral Method [42]. As the number of nodes used in calculating an optimal solution increases, it can significantly increase both the computational time necessary to develop a solution and the error between the optimal solution found using

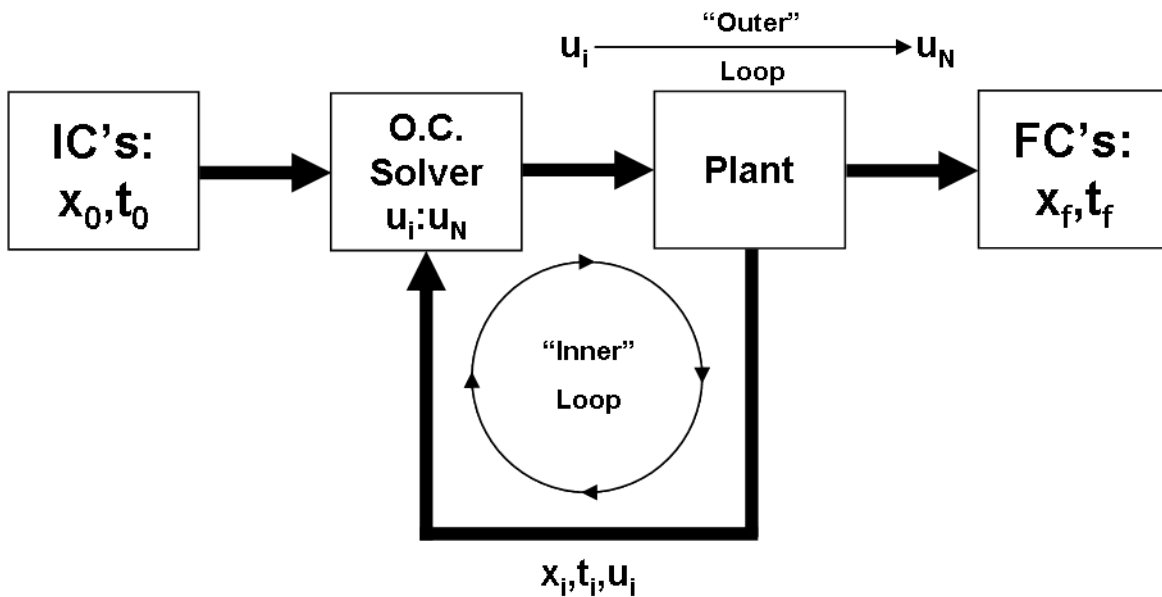


Figure 2.13: Real-Time Optimal Control - Control Loop

pseudospectral methods and the true analytical solution. In a generic optimal control problem, Ross *et al.* were able to demonstrate that by calculating an optimal solution using a fixed “small” set of nodes, then by implementing Bellman’s principle of optimality and recalculating the optimal solution using the same “small” set of nodes starting from intermediate points along the original optimal trajectory, the error in the final states of the optimal control solution could be significantly reduced. This enabled the calculation of near optimal solutions using very few nodes which significantly decreased computational time. This research further investigates RTOC problems similar to those performed by Ross, Fleming, and others, and examines new methods for implementing a RTOC controller on a satellite simulator.

2.4.4 Space Application of Optimal Control. Early investigations into optimizing the performance of controllers in space applications did not use the optimal control methods previously described in this section. In 1988, Wie *et al.* explored the use of a quaternion feedback regulator for eigenaxis rotational maneuvers of an asymmetric spacecraft. Since the early 1960’s, it was widely believed that with the exception of a series of a special cases that an eigenaxis rotation was the “optimal” satellite attitude maneuver. Their research established guidelines for gain selection, guaranteeing

that a quaternion feedback regulator will consistently provide a near-eigenaxis maneuver while also guaranteeing global stability using a Lyapunov function. They also solidified a control method for consistently producing geometrically smooth maneuvers that are globally stable in the face of parametric uncertainty, which is a highly desirable trait for spacecraft control.

By 1993, Bilimoria and Wie had proven that an eigenaxis rotation is not the time-optimal solution to a rapid rotational maneuver [8]. Their contrarian results, found using Pontryagin’s Minimum Principle, demonstrated that for an axisymmetric spacecraft the optimal control solution resembled a “bang-bang” control input. When this maneuver is observed from the inertial-frame, the spacecraft experiences what Bilimoria and Wie called geometrically “complex motion” that had a “significant nutational component.” With the advent of pseudospectral methods based optimal control solvers, researchers such as Ross and Fleming, have been able reproduce Billimoria and Wie’s results, and also show that the eigenaxis rotation is not the time optimal solution for asymmetric spacecraft as well. In this research, optimal non-eigenaxis maneuvers are compared against optimal eigenaxis maneuvers.

2.5 Summary

This chapter provided a literature review of topics relevant to this research. The first section reviewed spacecraft dynamics, followed by a historical overview of satellite simulators. The next section discussed spacecraft characterization methods. The last section explained optimal control theory (OCT) and how it has been used in related research.

III. System Architecture and Characterization

THIS chapter will first discuss SimSAT II’s hardware and software, and then provide an explanation of the control models used. The next section thoroughly examines the completion of SimSAT II’s manufacturing, integration, and hardware characterization, and the results of the hardware integration testing are presented. Because there are two distinct research efforts in this document, spacecraft characterization and optimal control, the test setup and results of spacecraft characterization are combined in this chapter. The test setup and results for the optimal control research are described in later chapters. The implementation of properly characterized hardware into SimSAT II’s control model is required for RTOC control.

3.1 Hardware

SimSAT II is AFIT’s 2nd generation spacecraft dynamics and control testbed that was designed and built by Roach, Rohe, and Welty between 2007 and 2008 [40]. It consists of three major hardware systems:

- a Ground Station personal computer (PC),
- a tri-axial Air Bearing, and
- the SimSAT II satellite simulator.

The most significant difference between SimSAT I and SimSAT II is the satellite’s structural architecture. SimSAT II’s structure is based on a “tabletop” configuration shown in Figure 3.1. Another key difference is that SimSAT II uses electric motors and propeller blades to simulate thrusters, as compared to SimSAT I’s cold-gas thrusters and reaction wheels. Roach *et al.* completed a detailed trade study to determine the hardware configuration for SimSAT II that can be found in [40] and [39].

3.1.1 Ground Station. The SimSAT II Ground Station is a custom built PC with a Windows XP 32-bit operating systems that runs MATLAB 6.5 and Simulink 5. The primary purpose of the Ground Station is to control SimSAT II through a Linksys 802.11 wireless link to the satellite simulator’s onboard Minibox PC. From the Ground

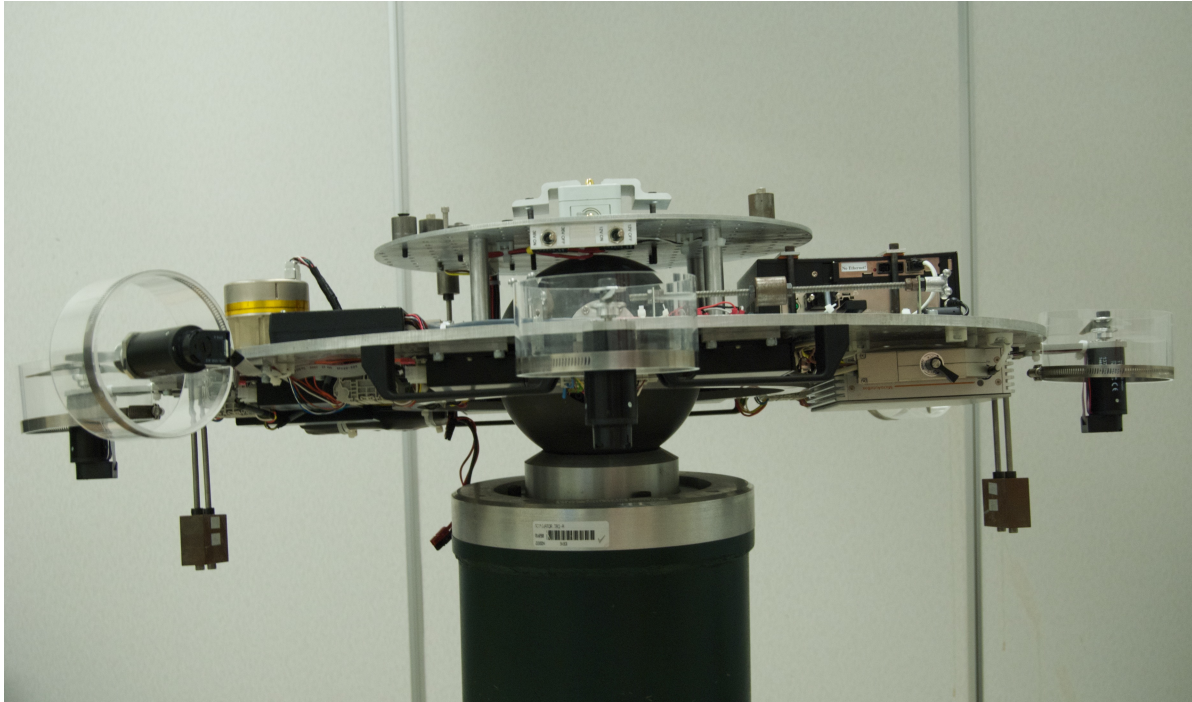


Figure 3.1: AFIT's SimSAT II simulated satellite system

Station, the operator logs onto the Mini-box PC through Windows Remote Desktop Connection (RDC) program native to XP. The other purpose of the Ground Station is to compute optimal control solutions using the DIDO PS-based optimal control software toolset that runs in MATLAB. After the optimal control solutions are computed, the Ground Station sends the control history to the the simulated satellite for implementation on the dSPACE MicroAutoBox. Detailed explanations of both the hardware and software interactions between the Ground Station and SimSAT II will be discussed further in Sections 3.2 and 3.3.

Under normal SimSAT II operations, both the RDC and MATLAB are run simultaneously, resulting in a prodigious amount of computer memory resources being consumed. During the course of initial testing on SimSAT II, the drain on computer memory resources from using both of these programs simultaneously brought about errors and delays in the spacecraft's attitude control system. As a result, the SimSAT II Ground Station utilized by Roach, Rohe, and Welty was upgraded from a standard dual processor desktop by 2008 standards to the recently developed quad-core *i7* processor from Intel. The upgraded Ground Station's components are presented in Table 3.1.

Table 3.1: SimSAT II Ground Station PC Components

Parameter	Value
Processor	Intel Core i7 2.66GHz Quad-Core Processor
Motherboard	BIOSTAR Intel X58 ATX Motherboard
Memory	4GB DDR3 SDRAM
Storage	240 GB
Video	Radeon x850 Series
Wireless	Linksys Wireless-G Broadband Router

3.1.2 Air Bearing. The SimSAT II simulated satellite uses the Space Electronics, Inc. Model SE-9791 Tri-Axis Spherical air bearing shown in Figure 3.2 to achieve nearly friction-free, with respect to the support mechanism, three DOF rotational motion, thus mimicking a torque-free environment. The air bearing consists of a heavy metallic pedestal with a spherically shaped cup on top and a corresponding ball bearing, both of which have finely machined surfaces. Air compressed to approximately 500 kPa is expelled from the cup through six outlet valves located along its surface, thus allowing the ball bearing, loaded up to 300 lbs., to float in the cup on top of an approximately 12 μm cushion of air. The structure of SimSAT II is attached to the ball bearing by two aluminum rods, as shown in the bottom view of SimSAT II in Figure 3.3. These aluminum rods and the tabletop structure of SimSAT II restrict it from rotating more than ± 25 degrees about both the 1 and 2 axis of the coordinate system defined in Figure 2.3, while the rotation about the 3 axis is unrestricted.

Table 3.2: Space Electronics, Inc. Tri-Axis Air Bearing

Parameter	Value	Unit
Ball Bearing Diameter	22.00	cm
Pedestal Cup Diameter	5.72	cm
Unloaded Ball Bearing Mass	19.05	kg
Maximum Loaded Ball Bearing Mass	136.08	kg
1-Axis Max Rotation Angle	± 25	deg
2-Axis Max Rotation Angle	± 25	deg
3-Axis Max Rotation Angle	-	-

3.1.3 SimSAT II Hardware. The SimSAT II simulated satellite is composed of a number of major subsystems including:



Figure 3.2: Space Electronics, Inc. Tri-Axis Spherical Air Bearing

- a mission-support computer subsystem,
 - Mini-Box PC
- a command and data handling (C&DH) subsystem,
 - dSPACE MicroAutoBox
- an attitude determination and control subsystem (ADCS),
 - LN-200 inertial measurement unit (IMU)
- and a propulsion subsystem.
 - Maxon EC Motors and Propellers

Each of the major subsystems are shown in Fig. 3.3. Detailed explanations of SimSAT II's major subsystems will be presented below. For more information about subsystems not included in the list found in Section 3.1.3, refer to [40] and [39].

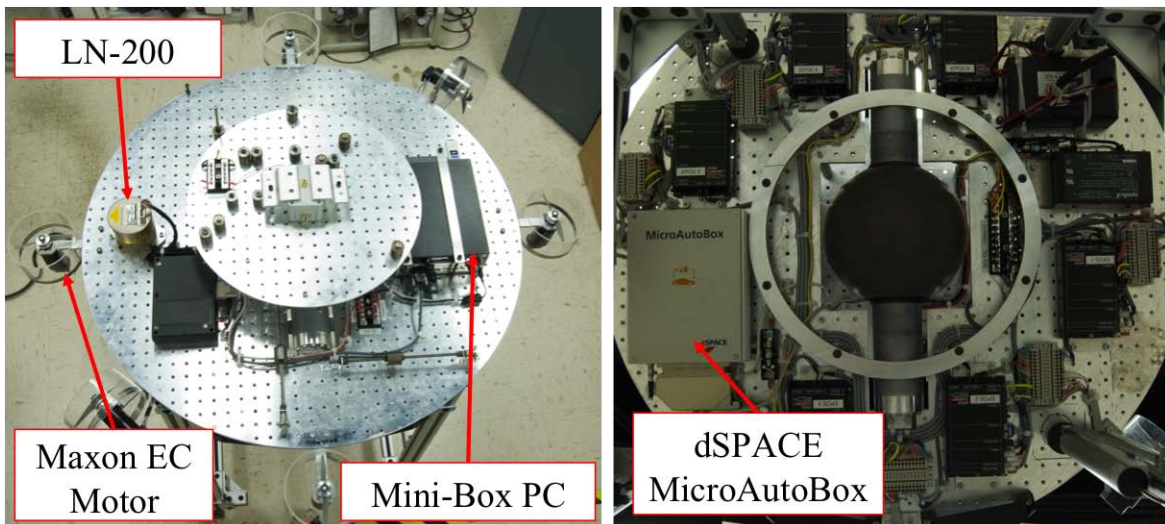


Figure 3.3: SimSAT II Top (Left) and Bottom (Right) View

3.1.3.1 Mini-box PC. SimSAT II's Mini-box PC is mounted to the top of the simulated satellite's bus plate as shown in Figure 3.4. In this research, the Mini-box PC performs two functions: providing wireless connectivity between SimSAT II and the Ground Station PC and transferring optimal control solutions from the Ground Station PC to the Mini-box PC. Through the wireless link, the operator on the Ground Station

PC is able to use the RDC to send telemetry commands to and receive state information from the Mini-Box PC. The Mini-box PC also serves as a hardwired interface to the dSPACE MicroAutoBox, also onboard SimSAT II. Using the software program, dSPACE ControlDesk, an operator is able to interact with SimSAT II's controller software in real-time. The detailed functions of the dSPACE MicroAutoBox are outlined in Section 3.1.3.3. More information about the software program dSPACE Control Desk is located in Section 3.2. The hardware components for the Mini-box PC are outlined in Table 3.3.

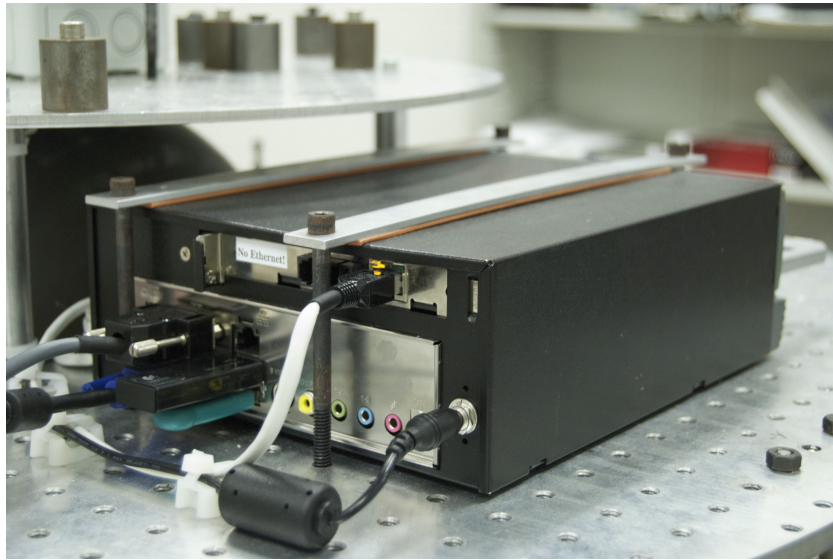


Figure 3.4: SimSAT II Mini-box PC

Table 3.3: SimSAT II Mini-box PC Components

Parameter	Value
Processor	x86 1500MHz
Motherboard	Jetway Hybrid MicroATX
Memory	1024 MB
Storage	40 GB
Wireless	Linksys Compact Wireless USB Network Adapter

3.1.3.2 LN-200 Fiber Optic Gyroscope. SimSAT II uses a Northrop Grumman LN-200 IMU pictured in Figure 3.5 for on-board attitude determination. The LN-200 consists of three fiber optic gyroscopes (FOG) and three accelerometers that are aligned with the principle axes of the device. FOGs consist of long coils of fiber optic

cable, up to several kilometers long. A laser diode is used to transmit light at either end of the cables. As the gyroscope rotates, light traveling around the axis of rotation will experience a slightly shorter path due to the Sagnac effect. A sensor also located at either end of the fiber optic coil is characterized to measure the effects that the shorter path has on the transmitted light. The system translates the measurement into a measured angular rotation rate. By aligning the principle axes of the IMU with the principle axes of SimSAT II, the angular rotation rates measured by the LN-200 IMU are the same as the angular rotation rates of the SimSAT II body frame. The accelerometers on the LN-200 were not used for this research. The LN-200 is a space-rated IMU that is currently used on AMRAAM medium range Air-to-Air missiles and the Mars Rovers. A detailed table of the LN-200's specifications can be seen in Table 3.4 [30].

Table 3.4: Northrop Grumman LN-200 IMU

Parameter	Value	Unit
Weight	700	g
Diameter	8.9	cm
Height	8.5	cm
Power Consumption	10	W
Bias Repeatability	1-10	/hr
Random Walk	0.04-0.1	$^{\circ}$ hr $^{\frac{1}{2}}$ power spectral density
Data Latency	<1	msec
Data Protocol	RS-485	-
Data Structure	-	Synchronous Data Link Control (SDLC)

The LN-200 cannot directly send its angular rotation rate measurements to the dSPACE MicroAutoBox. Instead, it must connect through a custom LN-200 interface board manufactured by SkEyes Unlimited Corporation. The LN-200 interface board converts the analog (SDLC) signal from the LN-200 Gyroscope into a 21 byte data packet that is transmitted over RS-232 Protocol to the dSPACE MicroAutobox.

3.1.3.3 dSPACE MicroAutobox. Real-time control of SimSAT II's hardware components is performed by a dSPACE MicroAutoBox mounted to the bottom of SimSAT II's bus plate shown in Figure 3.6. The dSPACE MicroAutoBox exclusively runs Simulink models that have been compiled into a C-coded executable format with a .ppc



Figure 3.5: Northrup Grumman LN-200 Fiber Optic Gyroscope

file extension using Simulink’s Real-Time Workshop toolbox. The dSPACE MicroAutoBox is connected directly to the Mini-box PC allowing the operator to manipulate control variables in real-time while the compiled Simulink control software is running on the MicroAutoBox. The operator can manipulate the control variables using either the dSPACE ControlDesk program or using a dSPACE designed MATLAB function library on the MATLAB command line. Relevant hardware characteristics data can be found in Table 3.5.

Table 3.5: DS1401/1501 dSPACE MicroAutoBox

Parameter	Value	Unit
Weight	2.15	kg
Width	182	mm
Length	192.6	mm
Height	50	mm
Power Consumption	30	W

3.1.3.4 *Maxon EC Motor and Propeller.* To simulate thrusters, SimSAT II utilizes six Maxon EC motor model #118895 with attached Landing Products LP05050



Figure 3.6: DS1401/1501 dSPACE MicroAutoBox

5x5 propellers and plexiglass safety cowling as seen in Figure 3.7. The Maxon motors, though originally designed for robotics and semiconductor equipment, are easily adapted for use as thruster simulators due to their high precision RPM rates at low electrical voltages and low vibrational output. The motors are digitally controlled through a Controller Area Network (CAN) interface that is integrated into the Simulink model being executed onboard the dSPACE MicroAutoBox. Two motors, rotating in opposite directions, are aligned with each of SimSAT II's principle axes. By spinning the motors in equal and opposite directions, the net angular momentum created by the spinning of the motor pairs was zero. Since thrust characteristics of these motors and propellers cannot be looked up in a table, the thrust produced by the propellers across the motors operating RPM speeds must be characterized. The thrust characterization efforts are discussed in detail in Section 3.4.3.

3.2 Software

Several software programs are integrated together to enable RTOC on SimSAT II. The first step in enabling RTOC is to create an accurate Simulink model of the SimSAT II system which has to provide a large number of basic functions including:

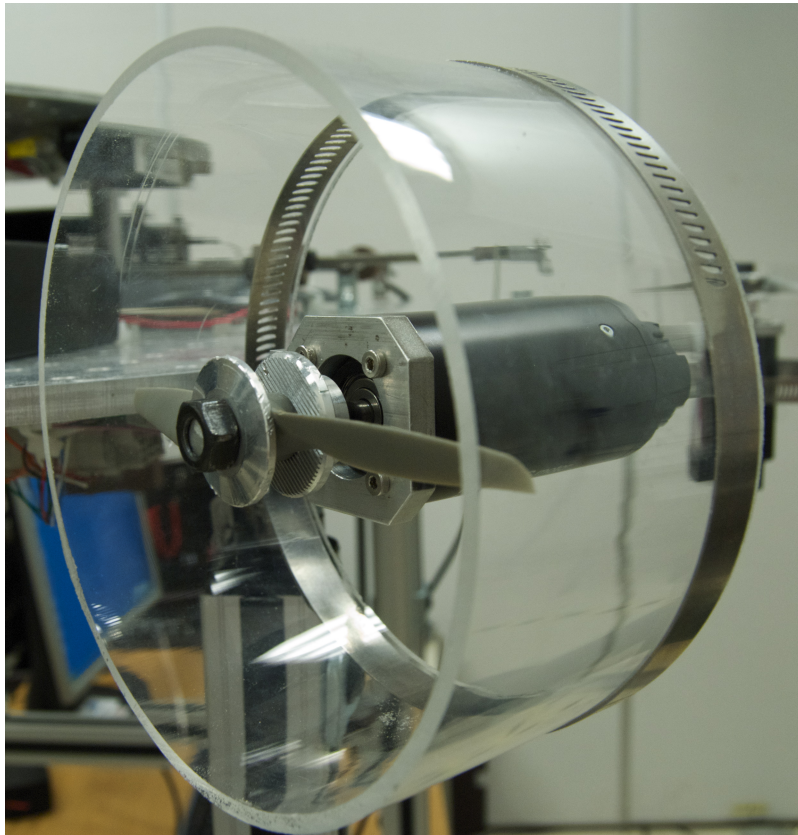


Figure 3.7: #118895 Maxon EC Motor and Landing Products LP05050 5x5 Propeller

- Providing Emergency Shutdown of Thrusters,
- Reading IMU Data and Calculating Current States,
- Calculating Desired Control States,
- Manipulating Spacecraft's Desired States,
- and Manipulating Thruster RPM Setting.

Figure 3.8 shows a top level outline of the Simulink controller model used on SimSAT II.

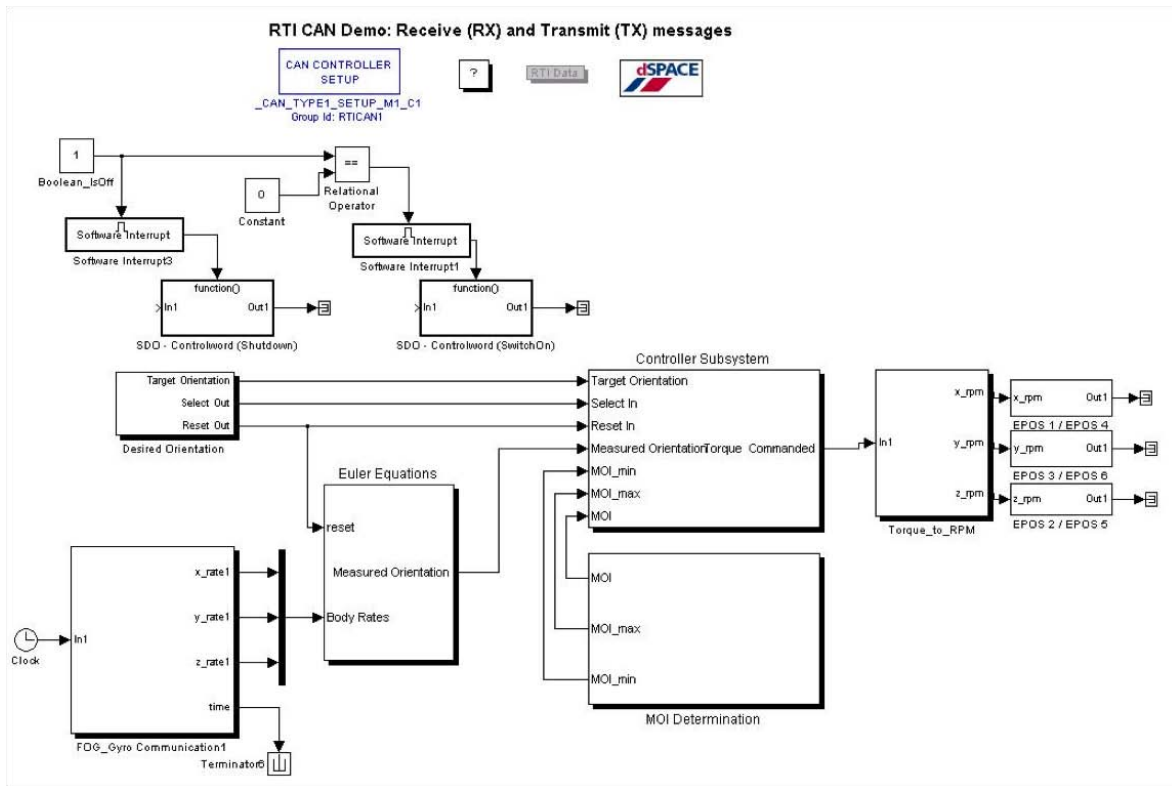


Figure 3.8: SimSAT II Top-level Simulink Controller Model

After completing the Simulink model, Simulink's Real Time Workshop toolbox is used to compile the model into C-based executable .PPC file. This executable file can be loaded on to the dSPACE MicroAutoBox using dSPACE ControlDesk program's graphical user interface (GUI). Once the .PPC file is loaded, a user customizable ControlDesk layout can be loaded, which allows the operator real-time monitoring and manipulation

of states, controls, and variable parameters. An example ControlDesk layout can be seen in Figure 3.9.

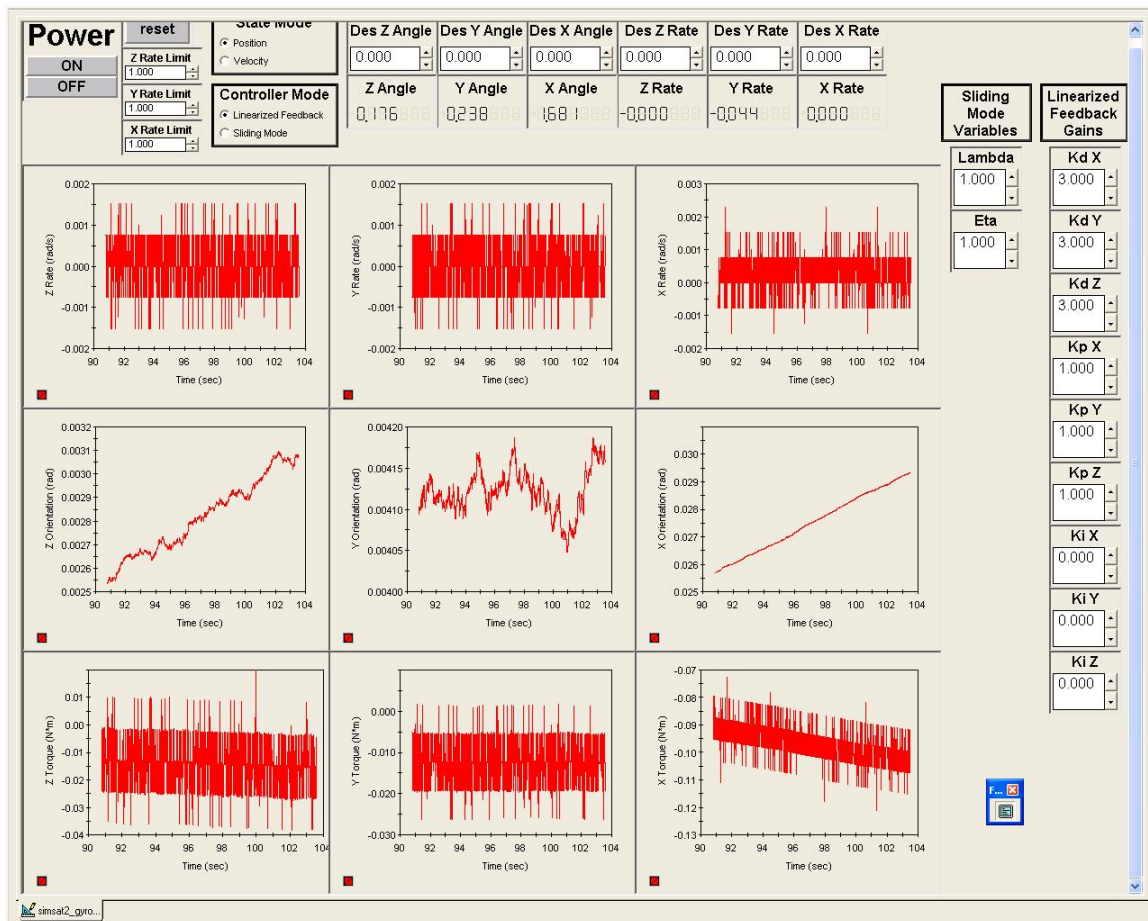


Figure 3.9: Example ControlDesk Layout for SimSAT II

Not all parameters inside the compiled Simulink model can be manipulated by the user inside the ControlDesk layout. Instead, a dSPACE MATLAB function library called “mlib” is used from the MATLAB command line. The “mlib” function library also allows near-simultaneous manipulation of variable parameters, while the ControlDesk layout only allows serial point and click manipulation. This enabled the operator to command state or control trajectories, which would otherwise be impossible to command using the ControlDesk layout alone. For example, the “mlib” function can be used inside a for-loop to enter a series of successive control commands from an optimal control solution trajectory.

3.2.1 DIDO. DIDO is a proprietary pseudospectral-based optimal control solver software package developed by Ross at the NPS [41]. DIDO allows a user to formulate an optimal control problem in a simple MATLAB m-code format. The DIDO approach of formulating optimal control problems into MATLAB scripts is much the same way you would approach a problem by hand. In this way, DIDO acts much like a “black-box” for calculating the solution to optimal control problems. The draw-back of using DIDO in a “black-box” manner is that users tend to blindly accept DIDO solutions as being correct. For more information about how to formulate a problem for DIDO, please refer to reference [41]. An example of an optimal control problem formulated for DIDO is shown in [41].

DIDO’s problem formulation approach provides the user with a large number of variables that can be modified to decrease computation time or improve accuracy. Results shown in Chapter V demonstrate that small changes in the formulation of the problem can result in drastically different solutions. While these solutions may appear correct at first glance, after checking the results through simulation, it becomes clear that the results are non-optimal and the problem will need to be reformulated. Therefore, DIDO calculated solutions to new optimal control problems should not be applied to systems without extensive verification of their optimality.

Because of DIDO’s sometimes high computational requirements, DIDO was run on the Ground Station to minimize the time it takes to compute a solution. Performance testing of the Minibox PC clearly verified that DIDO could not be run on the Minibox PC. After DIDO calculates the solution with the user defined number of nodes, the solution is saved as a .MAT file and sent via wireless link to the Mini-Box PC [41]. The mini-box PC then linearly interpolates the DIDO solution in MATLAB and discretizes the interpolated solution into 0.2 second intervals. This discretized set of controls is passed to the dSPACE MicroAutoBox using the “milb” function as discussed in Section 3.2. This process will be discussed in more detail in Chapter IV.

3.3 System Interfaces

SimSAT II's hardware and software architecture is highly complex, involving both wired and wireless interfaces between many pieces of hardware and software. SimSAT II's interface requirements and specifications can be found in reference [39] and [40], however, it is still difficult to visualize how each component physically or virtually interfaces with SimSAT II.

Figure 3.10 shows SimSAT II's hardware architecture, which includes both the major components explained in Section 3.1 and the minor components referenced in [39] and [40]. Wired interfaces are represented by solid lines and wireless interfaces are represented by dashed lines.

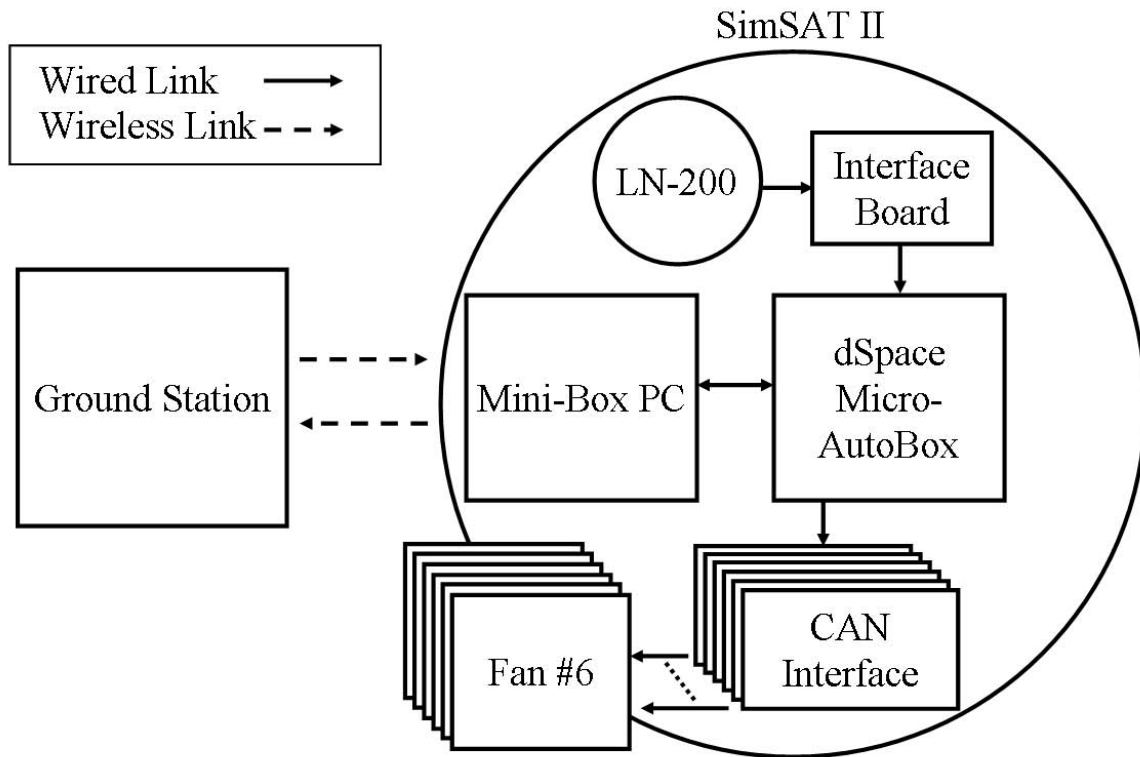


Figure 3.10: SimSAT II Hardware Architecture

Figure 3.11 shows SimSAT II's software architecture. This figure includes SimSAT II's major pieces of hardware and the relevant software being used. Hard-wired interfaces are represented by solid lines and wireless interfaces are represented by dashed lines, while

software only interfaces are represented by dash-dotted lines. Interface annotations show the file-type or the type of information being exchanged over the link.

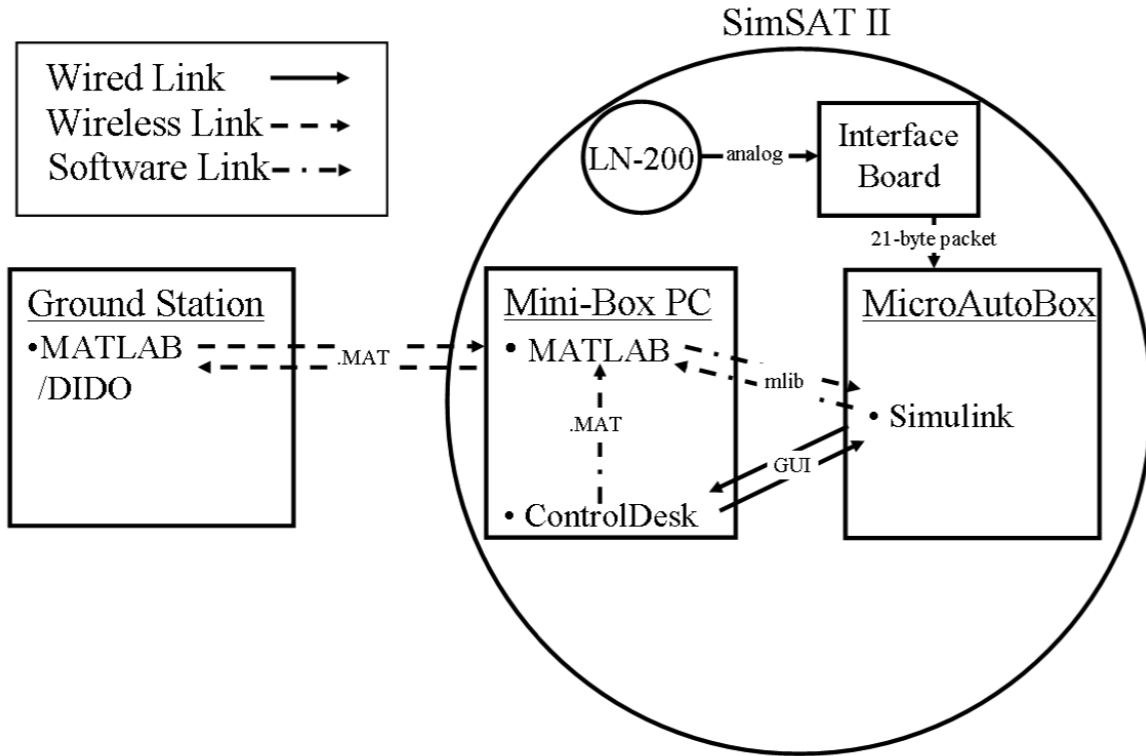


Figure 3.11: SimSAT II Software Architecture

3.4 System Characterization

While a majority of the manufacturing and integration was complete when this research effort started, several key components including the LN-200 Fiber Optic Gyroscope, the fan thrusters, and the Simulink controller model were not integrated. In order to enable RTOC, the following intermediate goals needed to be completed:

- Attach IMU to dSPACE MicroAutoBox and Integrate into Simulink Model,
- Filter Signal Spikes Found in IMU Data,
- Determine SimSAT II MOI Mass Properties,
- and Characterize Thrust Profile.

3.4.1 *Connecting LN-200 Fiber Optic Gyroscope.* In order to pass the measured angular rotation rate data sent by LN-200 IMU to the dSPACE MicroAutoBox, a RS-232 serial cable from the LN-200 Interface Board was hardwired into the dSPACE MicroAutoBox. The dSPACE MicroAutoBox uses a zero insertion force (ZIF) connector, so the standard RS-232 serial cable from the LN-200 Interface Board was modified to fit the MicroAutoBox’s ZIF connector. RS-232 protocol blocks were also added to the SimSAT II Simulink model that is run on the dSPACE MicroAutobox. The RS-232 blocks added to the Simulink model are shown in Figure 3.12.

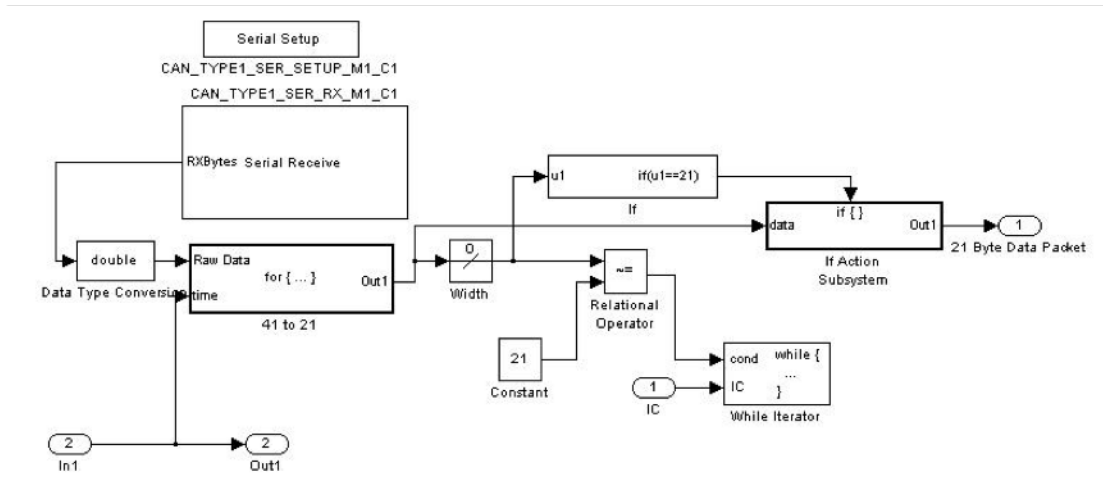


Figure 3.12: RS-232 Blocks in SimSAT II Simulink Control Model

3.4.1.1 *LN-200 Data Corruption.* Angular rate signals sent by the LN-200 can be displayed and stored using ControlDesk program on the Mini-Box PC. However, the signal displayed on ControlDesk showed sporadic, sharp increases or decreases in the signal’s magnitude, which will be referred to as “isolated gyro corruption,” as shown in Figure 3.13. These sharp increases and decreases in the LN-200’s signal are reminiscent of the isolated data corruption problems experienced by Hines on SimSAT I because SimSAT II uses the same LN-200 IMU and Interface Board as SimSAT I [30].

Hines recognized that no data corruption was experienced when the RS-232 cable from the gyro interface board was connected to a PC. Instead, data corruption only occurs when the IMU is connected to the dSPACE computer. By connecting the RS-232 cable to the serial port on a PC, one can run a MATLAB script that receives that

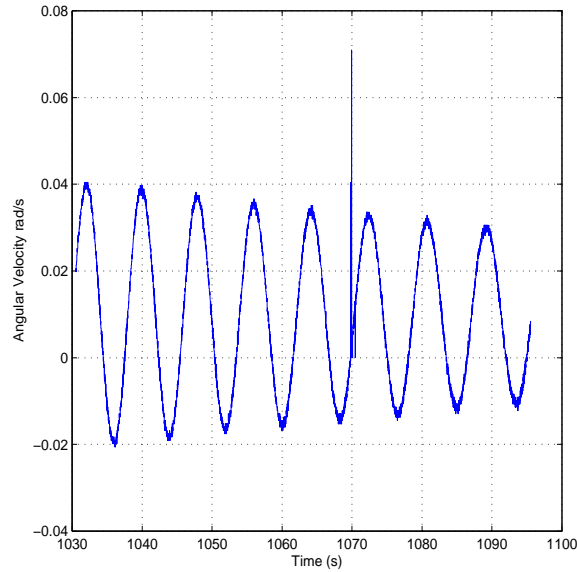


Figure 3.13: Example LN-200 Gyro Data Spike

data from the gyro and converts the digital signal to physical values much the same way it is accomplished using Simulink blocks. This test was repeated for the Gyro and interface card as installed on SimSat II, without receiving any data corruption. However, it should be noted that isolated gyro corruption generally occur while SimSAT II was spinning or rapidly changed its angular velocity. Since it is difficult to reproduce this extreme movement while also being physically connected to computer, it is possible that the isolated gyro corruption can also occur with a direct PC connect if the conditions are right for the corruption to occur.

Like Hines, a filter was implemented to replace corrupted data with data from previous steps. This filter did work for SimSAT I, but when applied on SimSAT II, it only worked under few sporadic circumstances or not at all. Therefore, new filters were created and tested in this research.

3.4.1.2 Digital Filtering. Figure 3.13 shows a typical measured signal from the LN-200 with a data corruption spike. Because the measured signals have a much longer wavelength than the data corruption, a low-pass digital communications filter would seem like an obvious first choice for eliminating the data corruption signal. A signal's frequency is inversely proportional to its wavelength, therefore, the corrupted

data signal must represent a high frequency signal since the wavelength of the corrupted data is small relative to the uncorrupted data. Under the assumption that corrupted LN-200 data is represented by a high frequency signal, a low-pass filter can be used to remove the corrupted data. Low-pass filters allow low frequencies to pass while significantly degrading the magnitude of high frequency content. The attenuating effect that a low-pass filter has on high frequency content is shown by the Bode Plot in Figure 3.14. Therefore, using a low-pass on the signal produced by the LN-200, the filter will treat the corrupted signal as high frequency content and reduce the sudden change in the the magnitude of the gyro data.

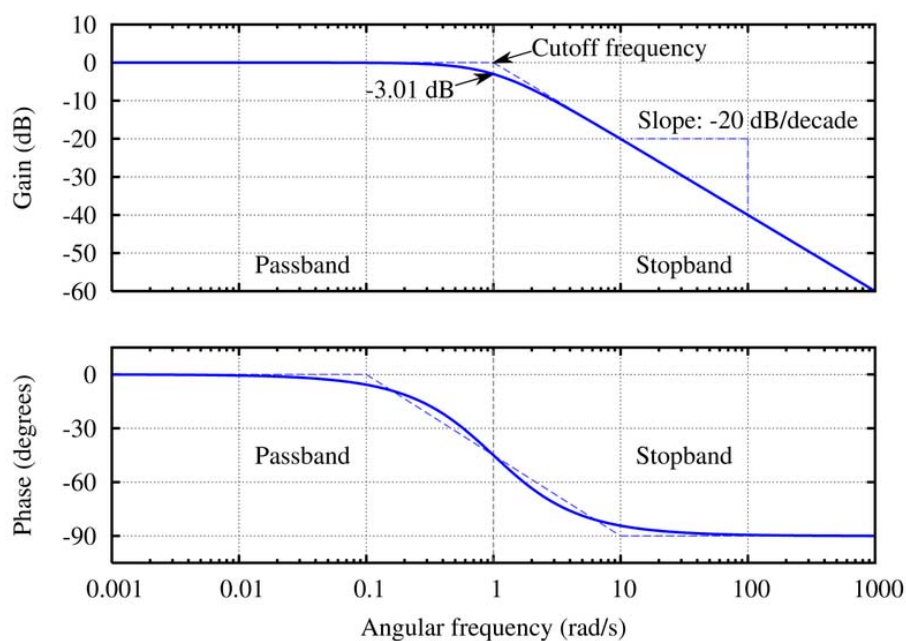


Figure 3.14: Bode Plot of a Butterworth Low-Pass Filter [2]

The drawback to using a low-pass filter is the phase lag that it creates in the signal. Figure 3.15 shows a variety low-pass filters that were tested on an isolated gyro spike recorded by the dSPACE MicroAutoBox. This significant reduction in the sudden magnitude change caused by the data corruption is clear; however, all of the digital filters cause noticeable phase lag in the signal even when no data corruption occurs at all. This phase lag will likely be more detrimental to the system than the data spike ever was. In closed-loop controllers, phase lag often results in instability or marginal stability in a

system because the controller is no longer synchronized with the system's actual physical movement.

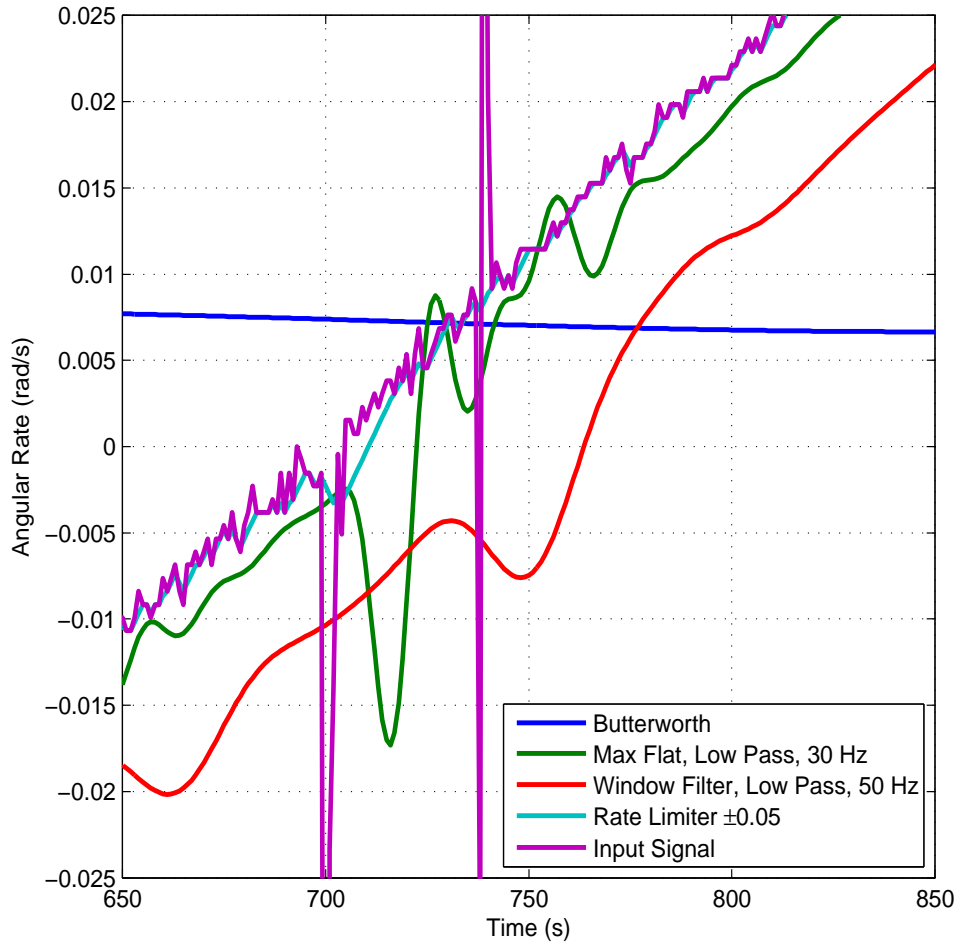


Figure 3.15: Digital Filter and Rate Limiting Techniques on Isolated Gyro Corruption

3.4.1.3 Rate Limiter. An alternative method for filtering the LN-200 isolated data corruption is a rate limiter. Since the LN-200 is measuring angular rotation rate, a rate limiter in this case would limit the change in angular rate between measurements. Figure 3.15 also shows a rate-limiter that was tested on an isolated gyro spike recorded by the dSPACE MicroAutoBox. The results of the rate limiter induce much less phase lag into system, when compared to low-pass filters.

The primary drawback to using a rate limiter is when SimSAT II has a change in angular rate faster than the maximum rate limit, then the actual change in angular rate would not be read by the system. The effects of using a rate limit that is too

constricting can have significant effects on a oscillating signal, such as what one would find on a spin-stabilized spacecraft. Figure 3.16 compares a harmonic signal with no rate limit, to the same harmonic signal where the rate limit is significantly lower than the angular acceleration of the spacecraft. The harmonic signal with the over-constrained rate limit tends to look like a saw-tooth wave, instead of sine wave.

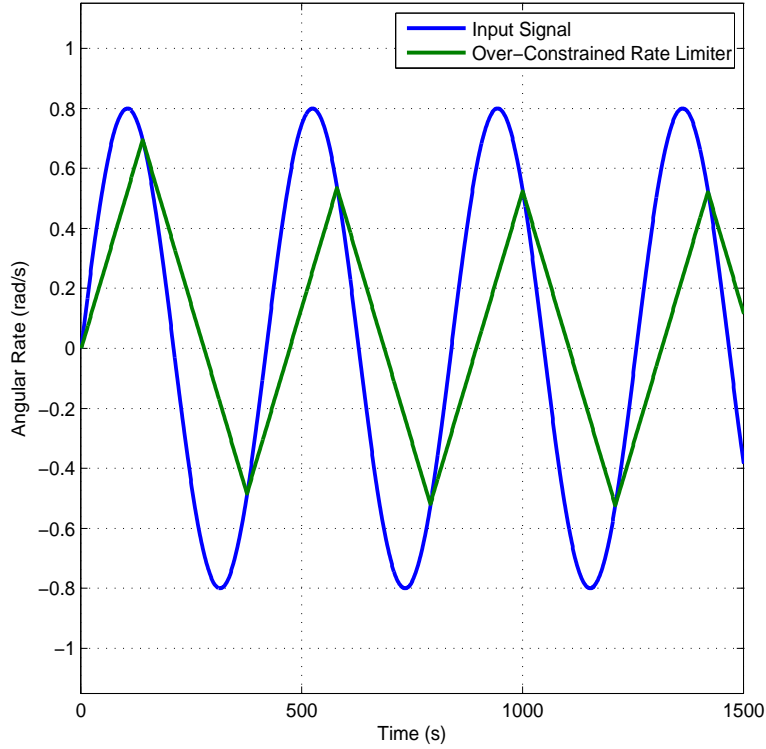


Figure 3.16: Effects of Over-Constrained Rate Limiting of Oscillating Signal

The smallest rate limit that should be used on SimSAT II is determined using the maximum torque produced by the thruster simulators, and SimSAT II’s MOI. Equation (2.20) can be rewritten as

$$\dot{\omega}_{bi} = \alpha = \frac{M}{I} \quad (3.1)$$

where α is the time rate of change of angular rate, M is applied torque, and I is the MOI about the axis of rotation. Using Equation (3.1), the maximum time rate of change in angular rate α is calculated for all three of the body axes, since the known maximum allowed torque generated by each set of fans is 220 mN-m and the MOI for SimSAT II

is 3.8, 3.2, and 5 kg m^2 about 1, 2, and 3 body frame axis, respectively. Therefore, the smallest rate limit that should be used is 0.058, 0.069, and 0.037 s^{-2} about the 1, 2, and 3 body frame axis, respectively. SimSAT II's current Simulink model uses a rate limit of 1 s^{-2} as shown in Figure 3.17.

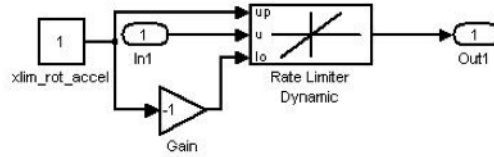


Figure 3.17: Rate Limiter Blocks in SimSAT II Simulink Control Model

3.4.2 MOI Mass Property. SolidWorks, computer aided drawing (CAD) tool, enabled Roach, Rohe, and Welty to calculate a fairly accurate estimation for SimSAT II's MOI matrix, Eq. (2.22), as seen in Table 3.6 [39]. Using an estimate of the mass properties for all of SimSAT II's individual components and their location on the spacecraft, Roach, Rohe and Welty used SolidWorks to determine the spacecraft's total MOI matrix [54]. Even with the tightest of manufacturing tolerances, there are usually differences between the mass properties of the design of a spacecraft, what is actually fabricated, and what the spacecraft experiences on orbit [32]. Roach, Rohe, and Welty noted during their thesis defense that all wiring and counter-weights used for balancing were not included in SimSAT II's fully loaded MOI estimation [40]. Therefore, it was necessary to perform simple characterizing tests to determine SimSAT II's actual MOI matrix.

Table 3.6: SimSAT II MOI Matrix - Fully Loaded

$I_{xx} = 4.474 \text{ kg m}^2$	$I_{yx} = 0.083 \text{ kg m}^2$	$I_{zx} = -0.006 \text{ kg m}^2$
$I_{xy} = 0.083 \text{ kg m}^2$	$I_{yy} = 4.133 \text{ kg m}^2$	$I_{zy} = 0.004 \text{ kg m}^2$
$I_{xz} = -0.006 \text{ kg m}^2$	$I_{yz} = 0.004 \text{ kg m}^2$	$I_{zz} = 6.786 \text{ kg m}^2$

3.4.2.1 Thruster Torque. To determine SimSAT II's actual MOI matrix, a known torque from the thruster simulators was applied to each of SimSAT II's principle

axes. This method uses several underlying assumptions to simplify Eq. (2.21). First, the POI for SimSAT II are assumed zero. This assumption is acceptable because the POI estimations in Table 3.6 are very small relative to the MOIs estimations. Another assumption is that ω^{bi} is zero for the axes where the known torque M is not being applied. Additionally, ω^{bi} is assumed to be small about the SimSAT II principle axis where the known torque M is applied, because the input M is small. Using these assumptions, the magnitude of the $\omega^{bi} \times I\omega^{bi}$ terms in Eq. (2.21) will be small, and can therefore be ignored. By rewriting Equation 2.21 as

$$I = \frac{\alpha}{M} = \frac{\dot{\omega}_{bi}}{M} \quad (3.2)$$

I can be determined because the known torque M produced by properly characterized thruster simulator and α can be calculated by estimating the slope of ω_{bi} about the SimSAT II principle axis where M is applied.

The thruster simulators are characterized by the thrust produced, but to use Eq. (3.2) the torque produced by the fans must be known. The torque M is calculated by each fan pair is

$$M = rT \quad (3.3)$$

where r is the sum of the distance from SimSAT II's CR to the center axis of both of the propellers in the fan pair, and T is the thrust produced by the fan pairs about the desired axis.

A MATLAB script using the “mlib” function library input a pattern of positive and negative thrust values into SimSAT II's simulink model. The pattern of positive and negative thrust inputs prevented SimSAT II from reaching its rotational limits as shown in Section 3.1.2. Inputting a known torque on SimSAT II using the fan pairs enabled multiple tests to determine SimSAT II's MOI matrix without physically touching the SimSAT II. This also allows for statistical analysis of the recorded data. Typical results of SimSAT II's determined MOI using this method is shown in Figure 3.18. The results in

Figure 3.18 show that SimSAT II's MOI is different when rotating in a positive direction than when rotating in a negative direction. Similar results were recorded for tests about all three principle axes of SimSAT II. Since it is not physically possible for a rigid body, such as SimSAT II, to have varying MOI parameters based on the direction of rotation, the error in the results is believed to be caused by an error in the thrust characterization of the thruster simulators. Therefore another method is needed for applying a known torque about SimSAT II's principle axes to determine SimSAT II's MOI matrix. The error in the thrust characterization is further discussed in Section 3.4.3.

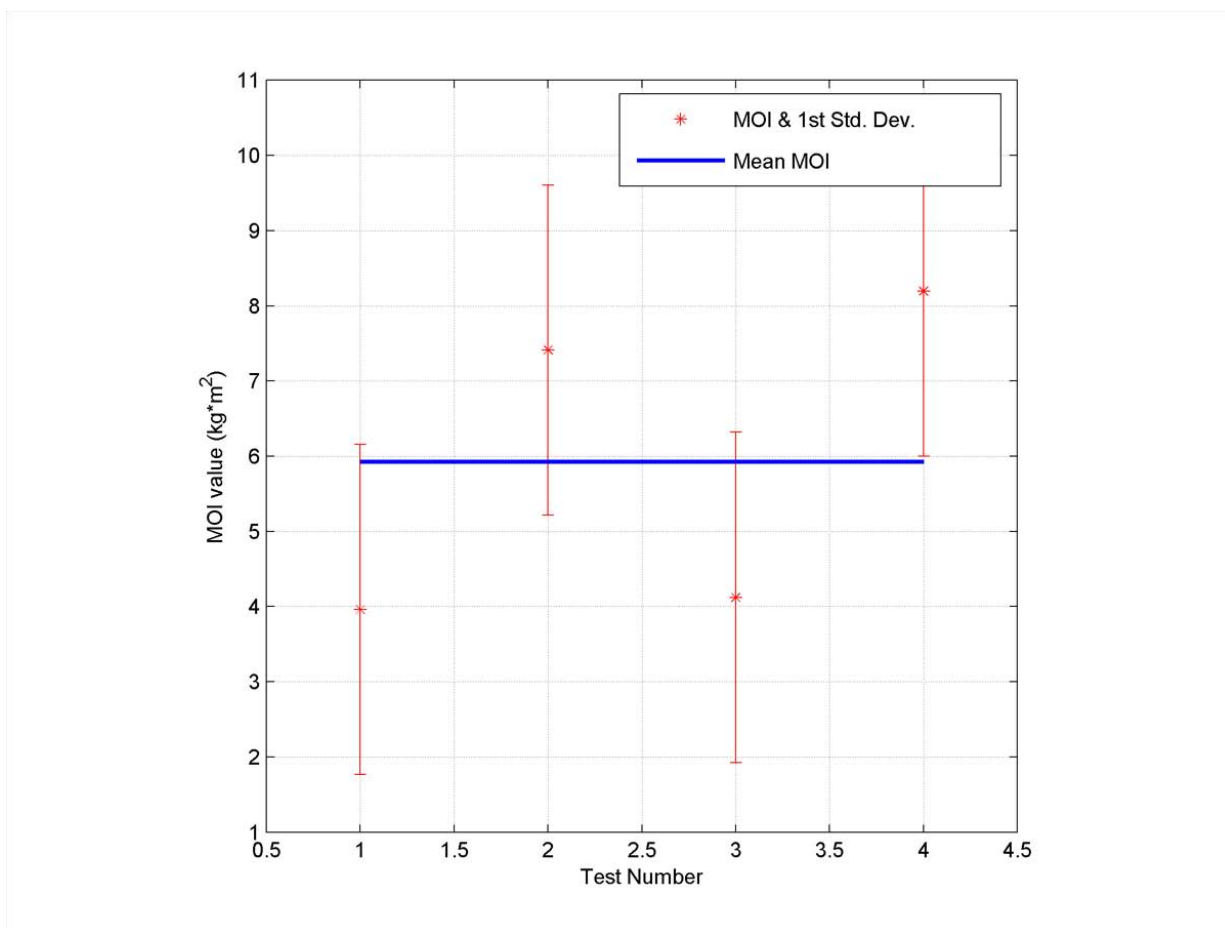


Figure 3.18: Initial MOI Estimation Using Fan Thruster Torque

3.4.2.2 Hanging Mass Torque. Another method for applying a known torque about SimSAT II's principle axes is to hang a mass from a string that results in a torque about one of SimSAT II's principle axes. The torque caused by the mass can be calculated from Eq. (3.3) where r is the distance from SimSAT II's CM to where

the string was attached, and T is simply the weight of the mass. Similar to the method discussed in Section 3.4.2.1, α is calculated by estimating the slope of ω_{bi} about the SimSAT II's principle axis where M is applied. The mass moment of inertia I about one axis is calculated using Eq. (3.2). By keeping SimSAT II's roll and pitch angles within $\pm 15^\circ$ of the inertial axes, there is $< 5\%$ error in the known torque due to cosine corrections. The cosine correction can clearly be seen as the difference between r and r' in Figure 3.19.

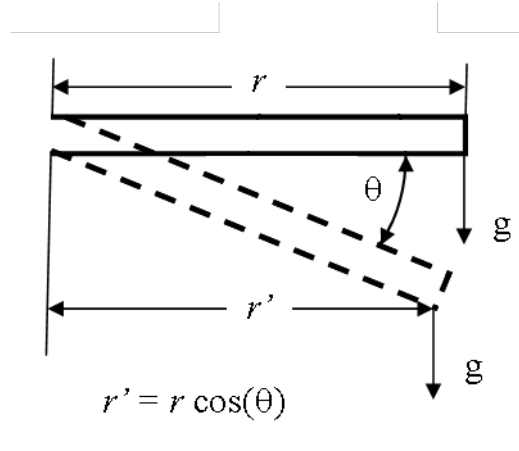


Figure 3.19: Error in Known Torque due to Cosine Losses

A hanging mass test was performed approximately 10 times for each of SimSAT II's principle axes. The average measured MOI for each of the principle axes is shown in Table 3.7. These MOI values are used in all calculations involving MOI values found in this research.

Table 3.7: Final SimSAT Measured MOI Values

Parameter	Value	Unit
1-axis MOI	3.8	kg-m ²
2-axis MOI	3.2	kg-m ²
3-axis MOI	5.0	kg-m ²

3.4.3 Simulated Thruster Characterization. Due to the fact that SimSAT II's propellers operate in a low Reynolds number Re regime, characterizing the Maxon EC motors and propellers is the only reliable way to determine a relationship between motor RPM and thrust. Accurately characterizing a system's control actuator, in this case

simulated thrusters, is vital to being able to use closed-loop control techniques. If the control actuator of a system is not producing the predicted control amount, as calculated by the system's control law, then new uncontrollable unknowns have been added to your system that put the system in jeopardy of becoming unstable. This research characterizes SimSAT II's simulated thrusters, or fans, by using the following three approaches:

- Static Thrust Characterization,
- Drag Equivalent Thrust Characterization, and
- Piecewise Angular Acceleration Estimation.

Each approach and their results will be described in this section.

The magnitude of control characterization accuracy required for control actuator is system dependent. For SimSAT II's near-RTOC controller, it was estimated that the steady state error in the angular rotational rate must be $<.01 \text{ s}^{-1}$. Using SimSat II's MOI about the 1, 2, and 3 principle axes is 3.8, 3.2, and 5 kg m^2 , respectively, and Eq. (2.20), then the fans accuracy to which the torque tolerance must be controlled is estimated to be 0.038, 0.032, and 0.05 Nm about the 1, 2, and 3 principle axes respectively.

3.4.3.1 Static Thrust Characterization. The fans were initially characterized using a static thrust approach. For this approach, a force gauge was attached to SimSAT II's perpendicular to the line defined by the fan and the axis of rotation being tested, as shown in Figure 3.20. The Maxon EC motors were commanded to $\pm 10,000$ RPM in 500 RPM increments. The force gauge reading was recorded at each 500 rpm increment. This process was repeated six times for the each fan pair, three times using positive RPM values and three times using negative RPM values. The thrust T measured for the positive and negative RPM values x were each fit to a second-order polynomial using MATLAB. The coefficients for the second order polynomial are located in Tables 3.8 and 3.9.

By assuming that the thrust being generated by the fans occurs only along the axle of the Maxon motors, the torque M created by each fan set from Eq. (3.3) can be calculated. The known distance r from the fan axles to the center of rotation of SimSAT

Table 3.8: Positive ω_i Static Thrust Fan Characterization Polynomial Coefficients

Fan #	$T = ax^2 + bx + c$		
	a	b	c
1	1.635×10^{-8}	-1.033×10^{-5}	8.544×10^{-3}
2	1.670×10^{-8}	-1.686×10^{-5}	9.437×10^{-3}
3	1.474×10^{-8}	-1.683×10^{-6}	6.529×10^{-3}
4	1.516×10^{-8}	-1.528×10^{-6}	3.086×10^{-3}
5	7.933×10^{-9}	-5.174×10^{-6}	-6.456×10^{-4}
6	1.495×10^{-8}	-5.697×10^{-6}	3.182×10^{-3}

Table 3.9: Negative ω_i Static Thrust Fan Characterization Polynomial Coefficients

Fan #	$T = ax^2 + bx + c$		
	a	b	c
1	-1.635×10^{-8}	-1.033×10^{-5}	-8.544×10^{-3}
2	-1.670×10^{-8}	-1.686×10^{-5}	-9.437×10^{-3}
3	-1.474×10^{-8}	-1.683×10^{-6}	-6.529×10^{-3}
4	-1.516×10^{-8}	-1.528×10^{-6}	-3.086×10^{-3}
5	-7.933×10^{-9}	-5.174×10^{-6}	6.456×10^{-4}
6	-1.495×10^{-8}	-5.697×10^{-6}	-3.182×10^{-3}

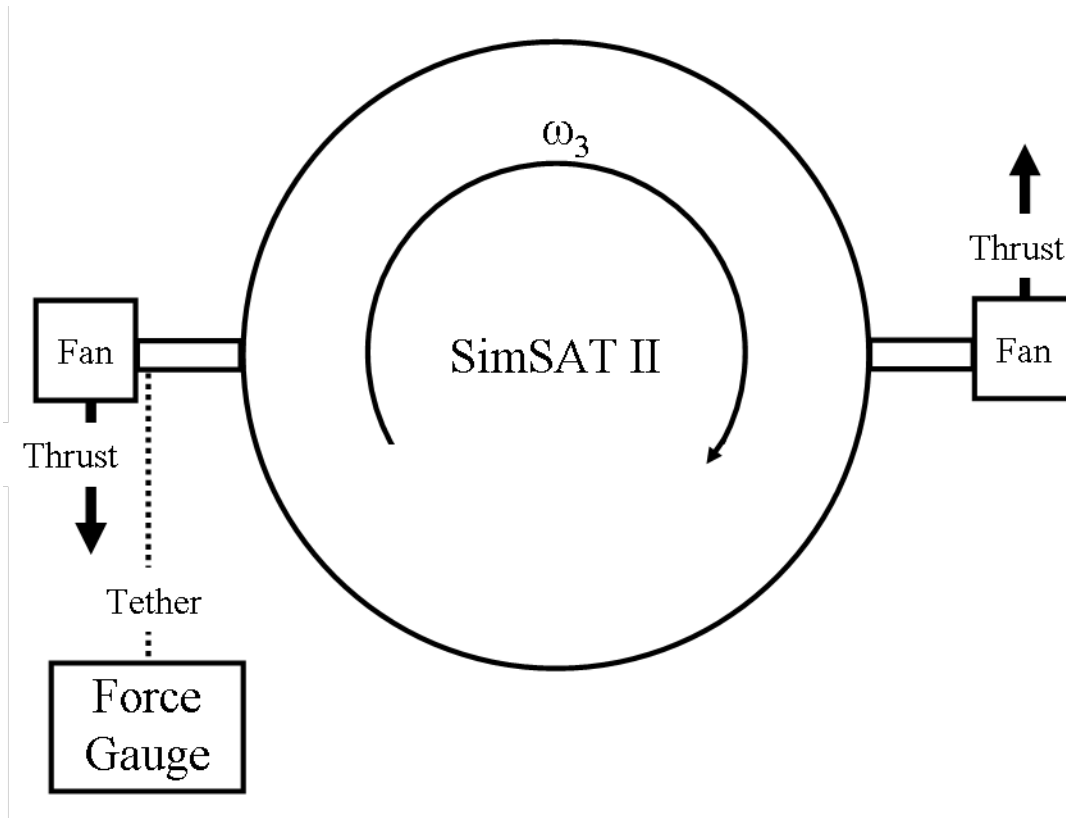


Figure 3.20: Static Thrust Characterization Setup

II is 114 cm, and the previously calculated second order polynomial characterizing the thrust T in terms of RPM. By replacing T in Eq. (3.3) with a second order polynomial, the quadratic equation

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \quad (3.4)$$

is used to solve for the polynomial's roots x using its coefficients a , b , and c from Tables 3.8 and 3.9. To find the fan RPM value x for a desired torque M , Eq (3.4) and Eq. (3.3) are combined as

$$x = \frac{-b \pm \sqrt{b^2 - 4a(c - \frac{M}{r})}}{2a}. \quad (3.5)$$

The quadratic equation has two solutions for x , so theoretically there are two RPM values that produce the desired torque. By evaluating the coefficients found in Tables 3.8 and 3.9, it shows that there is always a positive and a negative RPM value that produce the same amount torque. Based on SimSAT II's setup, the fans only produce positive torque spinning with positive RPMs and only produce negative torque spinning with negative RPMs. Therefore, if a desired torque is positive, only positive RPM value solutions to the quadratic equation are used, and vice versa for a desired negative torque.

Since all of the simulated thrusters use the same model propellers, motors, and controllers with similar internal gain settings, it was originally assumed that all of the motors would produce the same amount of thrust. The polynomial coefficients for the torque-to-RPM characterizations were implemented into the SimSAT II Simulink controller model for all three of its principle axes. Results from MOI characterization in Section 3.4.2 where the fans were used to input a known torque to the system demonstrated that SimSAT II's MOI must be different spinning in the positive direction than when it is spinning in the negative direction, as seen in Figure 3.18. This is not physically possible because SimSAT II is a rigid body and has constant MOI properties. Therefore, it was determined that motors must not be producing the same amount of thrust when using positive RPM inputs versus negative RPM inputs. Additionally, it was recognized that despite the similarity between all of the Maxon EC motors, propellers, and controller internal gain settings, each fan assembly is not going to produce the exact same amount of thrust for a given RPM.

Static characterization was performed for each of the six fans. The resulting second order polynomial fits for all six of the fans is found in Tables 3.8 and 3.9. These tests showed that motors 1, 2, 3, 4, and 6 all produced similar torque-to-RPM curves. On the other hand, Fan 5, which was aligned with the 3-axis, produced approximately half of the thrust that the other motors produced at corresponding RPM values. It was determined that the motor on Fan 5 needed to be replaced. Further testing of Fan 5 produced a torque-to-RPM curve similar to the other motors. The new torque-to-RPM curves were updated into the SimSAT II controller model, but when they were tested using the script to determine SimSAT II's MOIs, again the results showed that SimSAT II had different MOIs when spinning in the positive direction versus spinning in the negative direction.

These results led us to the conclusion that static thrust characterization of SimSAT II's simulated thrusters does not produce accurate torque-to-RPM curves. It is believed that the reason why static thrust characterization did not produce accurate results was due to the fact that a propeller's performance changes as a function of freestream velocity, but testing at static conditions does not take these changes into account. Research has shown that as the Reynolds number of the flow decreases, the thrust performance of the propellers will become more inconsistent [36]. Additionally, the accuracy of this test setup relied heavily on aligning the thrust vector from the motors with the force gauge measuring the thrust. Since these test were not performed by fully restricting off-axis movement, it is possible that some of thrust produce by the propellers was not fully measured by the force gauge. Another source of error could have been the force gauge itself, since its accuracy was limited to ± 0.05 mN.

3.4.3.2 Drag Equivalent Thrust Characterization. The second approach taken to characterize the thrust was to measure the drag on SimSAT II as it is slowed from an initial rotation rate. This was accomplished by using the fans to achieve an initial rotation rate around the 3-axis of approximately 6 rad/s. The fans were then turned off, and the angular rotation rate of SimSAT II was sampled approximately 20 times a second while the angular rotation rate slowed to zero due to the air drag torque. Since air drag torque may differ between positive and negative rotation directions due

to SimSAT II's asymmetric shape, this test was performed and measured using both positive and negative angular rotation rates.

With an estimate for the mass MOI about SimSAT II's 3-axis and using Eq. (2.20), the torque induced on SimSAT II is estimated from air drag alone. The angular acceleration α is estimated from the measured angular rotation rate data by estimating a derivative. One cannot directly calculate the discrete derivative from the measured IMU data because the angular rotation rate fluctuates rapidly between samples. If one were to compute a discrete derivative directly from this data set, the results would be discontinuous jumps as shown in Figure 3.21. This direct approach will not provide an accurate measure of angular acceleration. To alleviate this problem, the angular rotation rate data was fit with a 15th order polynomial using MATLAB's POLYFIT command. A 15th order polynomial was used because it was the lowest order polynomial that provided the closest fit to the measured data. Once the data is smoothed to a 15th order polynomial, one can easily take a derivative from the polynomial fit.

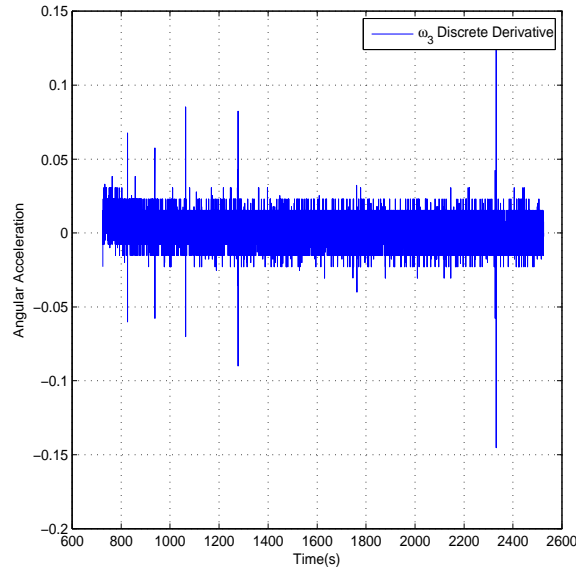


Figure 3.21: Discrete Derivative of Measured Angular Rate

With an estimate for the torque induced by drag as a function of angular rotation rates ranging from -6 to $+6$ rad/s, one can apply torque with the fans until SimSAT II spins at a constant rate about the 3-axis. When SimSAT II is spinning at a constant rate, it means that the torque applied by the fans is now equal to the torque created by air drag

at said angular rotation rate. Since the 3-axis thrusters are spinning at a constant rate for a constant torque, a torque-to-RPM relationship can now be determined. MATLAB was used to estimate the polynomial curve representing this relationship for all of the ± 500 RPM increments tested.

To test this torque-to-RPM relationship, a desired torque was commanded to SimSAT II's fans. By measuring the resulting angular rotation rate over a small time, one can linearly approximate the slope of the angular rotation rate to be equivalent to the angular acceleration experienced by SimSAT II. Using Eq. (3.1) and an estimate for SimSAT II's MOI, an estimate is formed for what SimSAT II's angular acceleration should be for a commanded torque. The measured angular acceleration was approximately 200% of the estimated angular acceleration for several commanded torque values. It was determined that the source of error for the second torque-to-RPM relationship came from the difficulty is SimSAT II achieving a constant angular rotation rate due to the fact that rotations about the 1-axis and 2-axis were not restricted. As a result, polynomial curve representing the torque-to-RPM relationship underestimated the amount of torque being produced by the thruster simulators.

3.4.3.3 Piecewise Angular Acceleration Estimation. A third approach used to characterize SimSAT II's simulated thrusters set the RPM of the fans to fixed ± 500 RPM increment and let SimSAT II spin from rest to approximately ± 1.2 rad/s. For RPM increments less than $\pm 1,500$ RPM, SimSAT II could not achieve ± 1.2 rad/s. Therefore measurements were taken from rest until SimSAT II was accelerating less than 0.01 rad/s². While SimSAT II was spinning, the angular rotation rate of SimSAT II was measured approximately 40 times per second by the LN-200 IMU. After SimSAT II achieved approximately ± 1.2 rad s⁻¹, the measured angular rate data was divided into 10 equal sized intervals. The slope of the angular rotation rate was taken for each interval as shown in Figure 3.22. The acceleration due to drag at the average angular rotation rate of each interval was subtracted from measured angular acceleration. Using Eq. (2.20), the measured angular acceleration, and SimSAT II's 3-axis MOI, the torque produced for each interval was estimated.

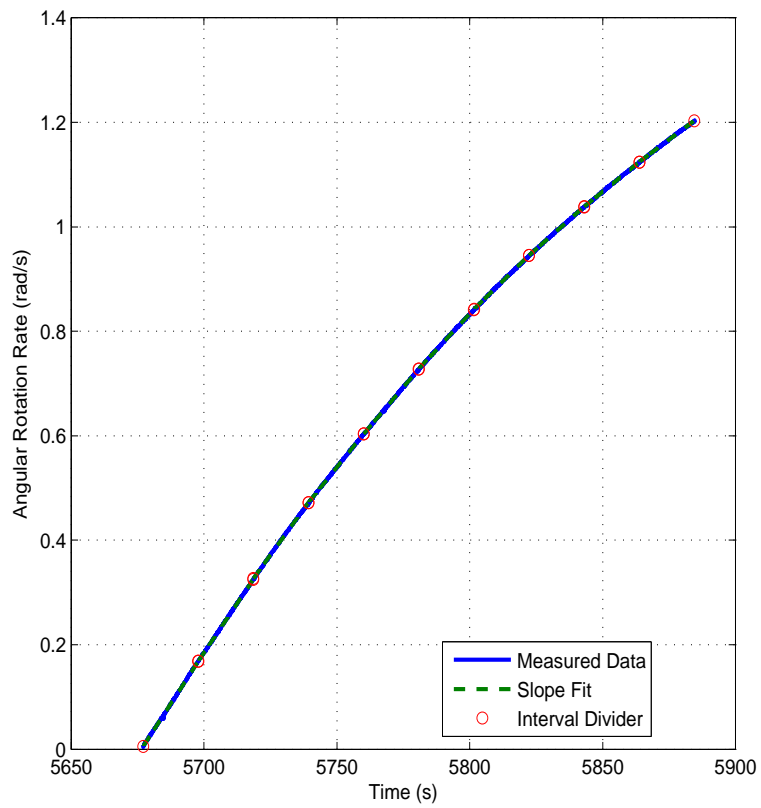


Figure 3.22: Angular Acceleration Slope Intervals for 3-Axis Thrusters at 2500 RPM

A minimum, mean, and maximum torque-to-RPM relationship was formed by linearly interpolating between the minimum, mean, and maximum, torque values recorded for each 500 RPM interval. Each interpolated curve, was tested for accuracy by inputting a desired torque, and measuring the actual angular acceleration produced versus an estimated angular acceleration based on the torque-to-RPM relationship. When the maximum torque-to-RPM relationship was used to produce a desired angular acceleration for the system, it produced a measured angular acceleration that more closely matched the theoretical results than the minimum or mean relationships. Therefore the maximum torque-to-RPM relationship was implemented into SimSAT II's Simulink model and used for all further controller testing.

The result of all three thruster characterization methods are compared in Figure 3.23. It is clear from this figure that the first static characterization approach overestimated the torque produced by the fans because it did not account for performance characteristics of the fans changing with freestream velocity. The second approach, drag equivalent thrust characterization, underestimated the torque produced because SimSAT II could not achieve a constant angular rotation rates due to the rotation about the 1-axis and 2-axis not being restricted. The third approach, piecewise angular acceleration estimation, yielded the best results because it is simpler in design than the other two methods and it accounted for the performance characteristics of the fans changing with respect to the freestream velocity. Despite producing better results, the piecewise approach still had some error because SimSAT II's rotation about the 1-axis and 2-axis was not restricted. Therefore future fan characterization testing should be conducted in an environment where rotation is restricted for the axes not being tested.

3.5 Control Models

The SimSAT II simulated satellite uses a Simulink control model to interact with all of its major subsystems components. The Simulink control model is executed on the dSPACE MicroAutoBox which serves as the hub for all of SimSAT II's control activity. The top-level view of SimSAT II's control model is shown in Figure 3.24. The major functions performed by the control model are as follows:

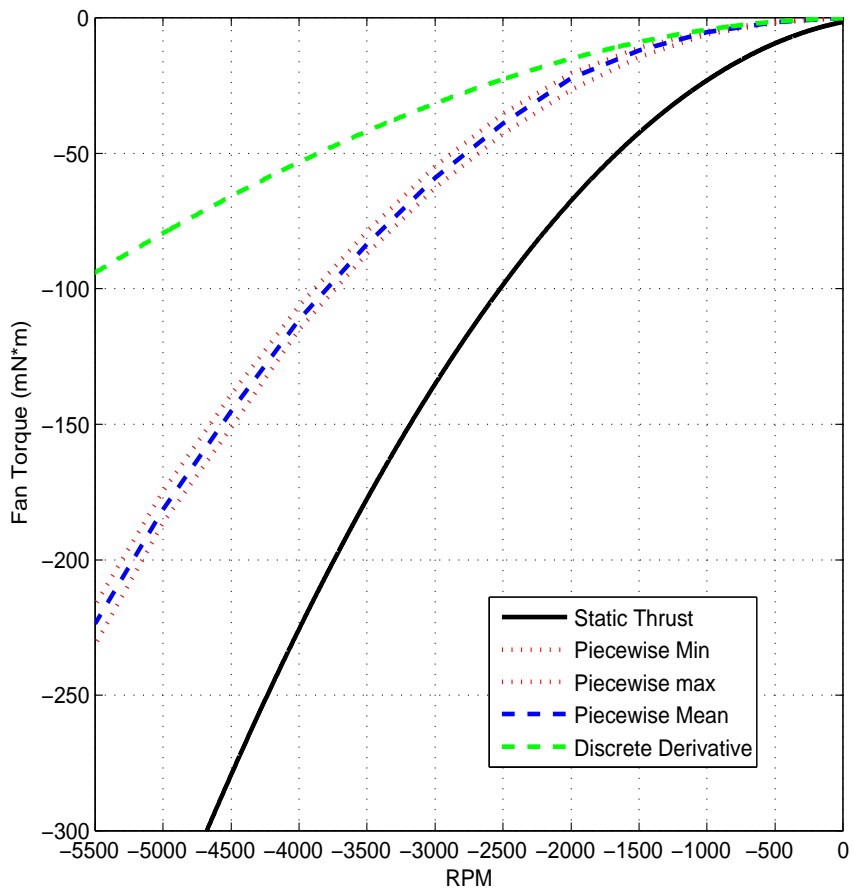


Figure 3.23: Comparison of Thruster Characterization Results

- Receives Measured Angular Rotation Rates from LN-200 IMU,
- Commands RPM Inputs to the Fans,
- Receives and Executes External Control Commands, and
- Performs Closed Loop Control Calculations (PID controller only).

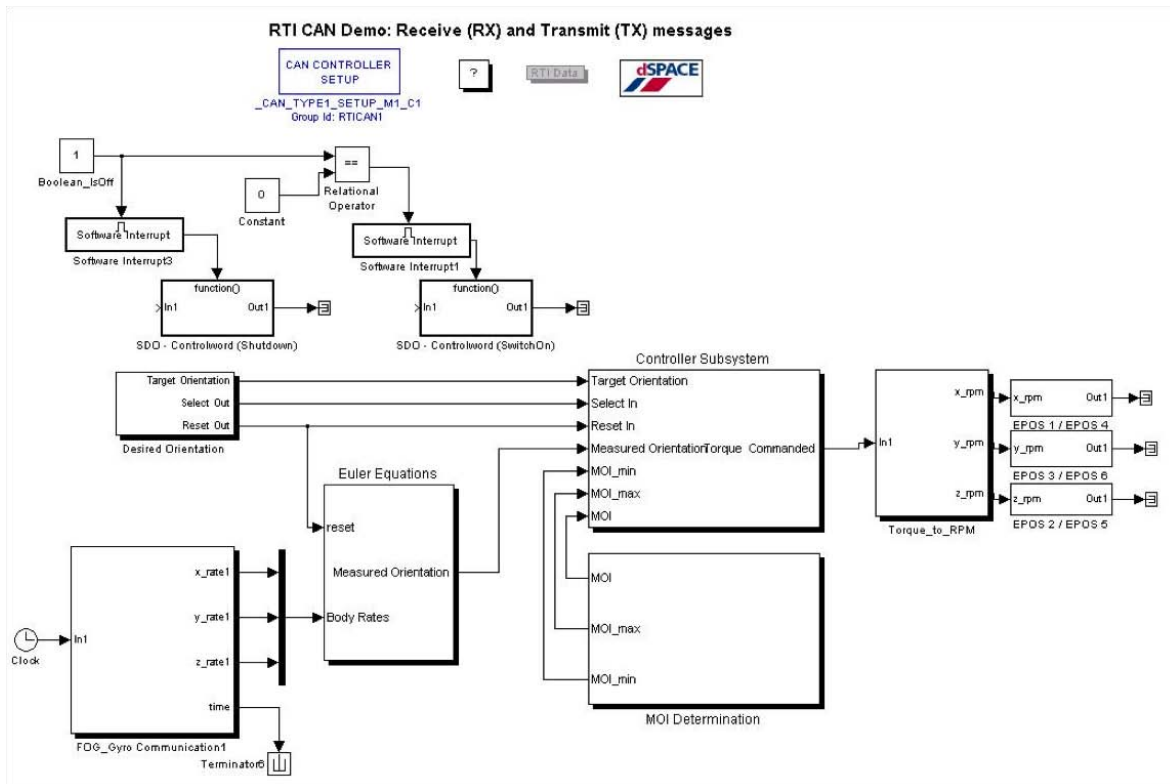


Figure 3.24: Top-Level View of SimSAT II Control Model

The LN-200 Interface Board asynchronously sends 21-byte packets containing body-frame angular rotation rate information to the dSPACE MicroAutoBox 400 times per second using the standard RS-232 serial protocol. Customized Simulink blocks created by dSPACE are included in SimSAT II's controller model to allow the dSPACE MicroAutoBox to read packets using the RS-232 protocols. SimSAT II's control model samples data from the LN-200 Interface Board 41-bytes at a time, and searches for one single continuous 21-byte packet containing the angular rotation rate information within the larger 41-byte sample. This approach effectively reduces the LN-200 IMU's cycle time from 400 Hz to less than 200 Hz. The 21-byte packets are received in binary from the LN-200 Interface Board and converted to base ten for use in future control calculations.

The Simulink model transforms the angular rotation rate data from the body-frame and discretely integrates it to determine SimSAT II's angular position in Euler Angles.

SimSAT II's simulated thrusters are digitally controlled using a CAN interface to the dSPACE MicroAutoBox. Custom Simulink blocks supplied by dSPACE are included in SimSAT II's controller model to allow the dSPACE MicroAutoBox to individually send RPM commands to each of the simulated thrusters over the CAN protocol. Desired RPM or torque commands for the simulated thrusters can be entered directly into SimSAT II's control model. Desired torque commands are converted to a RPM value using the torque-to-RPM relationship developed during the thruster characterization. To meet laboratory safety goals, SimSAT II's Simulink control model also enables emergency shutdown of the simulated thrusters.

3.5.1 PID Closed Loop Controller. A linearized Proportional-Integral-Derivative (PID) state feedback controller model was implemented on the original SIMSAT by Hines. This PID controller model was adapted and implemented on SimSAT II. The PID controller was only used in this research to demonstrate SimSAT II's ability to slew and maintain a commanded attitude. Using this PID controller as is, SimSAT II cannot slew more than $\pm 60^\circ$ about the 3-axis without going unstable. The maximum rotation angle of $\pm 25^\circ$ about the 1-axis and the 2-axis prevented SimSAT II from going unstable when slewing solely about these two axes. No further modifications to the PID controller were attempted for this research.

3.5.2 Optimal Control Implementation. Application of the optimal controller is only possible through the dSPACE developed "mlib" function library. This "mlib" function library enables reading and writing of all Simulink model variables while the .PPC executable file version of the model is running on the dSPACE MicroAutoBox. Optimal control solutions are composed of the optimal state trajectory and the required inputs torque from SimSAT II's fans to achieve the optimal state trajectory. To command SimSAT II's fans to generate the required input torque to achieve the optimal state trajectory, the "mlib" function continuously writes and rewrites the desired torque value in the Simulink model.

Initial testing has shown that the “mlib” function takes an inconsistent amount of time to execute. This results in a varying amount of delay between the time the control was commanded and the time when it was executed. To normalize the delay for control inputs, a pause function was implemented to force the “mlib” function to update control commands every 0.2 seconds. Figure 3.25 shows the mean and first standard deviation for the “mlib” command delay both with and without the use of the pause function. The nodal spacing of optimal control solution is not equally partitioned in time because DIDO uses LGL points to calculate an optimal control solution. Therefore, the DIDO solution is linearly interpolated into 0.2 second intervals to match the desired control update cycle time.

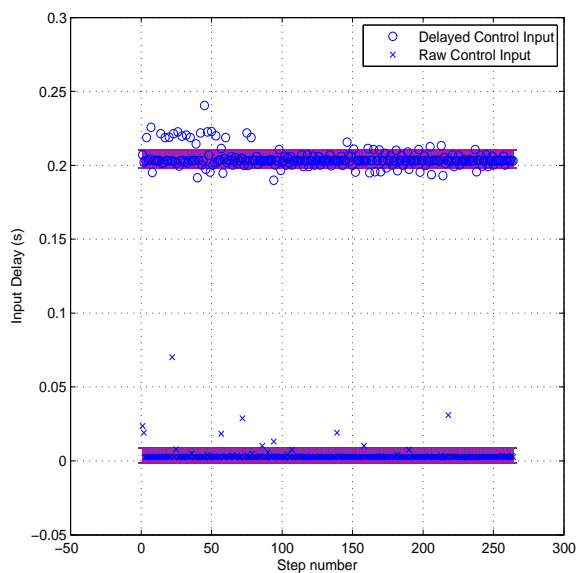


Figure 3.25: “mlib” Function Command Delay

The “mlib” function is used to read SimSAT II’s current states for angular rotation rate and angular position only during a RTOC maneuver. The recorded state information is then sent to the Ground Station PC to use for computing the next optimal control solution using DIDO.

3.6 Summary

This chapter first discussed SimSAT II’s hardware and software, and then provided an explanation of the control models used. The next section thoroughly examined the

completion of SimSAT II's manufacturing, integration, and hardware characterization, and the results of the hardware integration testing were presented. Because there are two distinct research efforts in this document, spacecraft characterization and optimal control, the test setup and results of spacecraft characterization were combined in this chapter. The test setup and results for the optimal control research are described in later chapters. The implementation of properly characterized hardware into SimSAT II's control model was required to implement a RTOC controller on SimSAT II's hardware.

IV. Real-Time Optimal Control

THIS chapter will first outline a fixed initial time, fixed initial state, fixed final state, free final time optimal control problem which includes SimSAT II's system dynamical constraints. The next section examines the approaches taken to implement OCT solutions on SimSAT II, followed by a thorough examination of the approaches taken to implement near-RTOC control on SimSAT II. The final section describes the approaches taken to model SimSAT II's traditional optimal control and RTOC controller.

4.1 *SimSAT II Optimal Control Problem Definition*

Fixed initial time, fixed initial state, fixed final state, free final time optimal control maneuvers for slewing SimSAT II are evaluated in this research. Since the goal is to complete a slew maneuver in minimum time, the cost function is simply defined as

$$J = \phi(t_f, \mathbf{x}_f) = t_f, \tag{4.1}$$

where the time t_f is an unconstrained variable. The Lagrange integral cost function $L(t, \mathbf{x}, \mathbf{u})$, as shown in Eq. (2.58), is zero.

Each maneuver starts from initial time

$$t_0 = 0, \tag{4.2}$$

with the initial conditions

$$\mathbf{x}_0 = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4.3}$$

where the quaternion states q_i and the angular rotation rates ω_i represent the rotational parameters of SimSAT II's body-fixed frame with respect to the inertial frame. The initial conditions given in Equations (4.2) and (4.3) define SimSAT II starting each maneuver from rest with its body-frame axes aligned with the inertial frame of reference. Each maneuver must also satisfy the final condition constraint vector

$$\psi(t_f, \mathbf{x}_f) = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sin(\theta/2) \\ \cos(\theta/2) \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.4)$$

at the unconstrained final time t_f . The angle θ represents a desired rotation from the 3-axis of the inertial frame with respect to the 3-axis of SimSAT II's body-fixed reference frame. The state values calculated using Eq. (4.4) define that SimSAT II will finish each maneuver at rest with its body-frame axes aligned with the axes defined by the desired the quaternion parameters. The set of desired quaternion parameters defined in Eq. (4.4) is restricted to a yaw axis rotation only, however this does not prevent SimSAT II from rotating about the roll and pitch axes during the maneuver. Any set of desired quaternion parameters using a Roll-Pitch-Yaw Euler Angle set can be calculated using Eq. (2.50).

SimSAT II's EOMs are defined by

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & \frac{1}{2}\omega_3 & -\frac{1}{2}\omega_2 & \frac{1}{2}\omega_1 & 0 & 0 & 0 \\ -\frac{1}{2}\omega_3 & 0 & \frac{1}{2}\omega_1 & \frac{1}{2}\omega_2 & 0 & 0 & 0 \\ \frac{1}{2}\omega_2 & -\frac{1}{2}\omega_1 & 0 & \frac{1}{2}\omega_3 & 0 & 0 & 0 \\ -\frac{1}{2}\omega_1 & -\frac{1}{2}\omega_2 & \frac{1}{2}\omega_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{B-C}{A}\omega_2 \\ 0 & 0 & 0 & 0 & \frac{C-A}{B}\omega_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{A-B}{C}\omega_1 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{A} & 0 & 0 \\ 0 & \frac{1}{B} & 0 \\ 0 & 0 & \frac{1}{C} \end{bmatrix} \mathbf{u}, \quad (4.5)$$

or rewritten in function form as

$$\begin{aligned} \dot{q}_1 &= \frac{1}{2}(\omega_3 q_2 - \omega_2 q_3 + \omega_1 q_4) \\ \dot{q}_2 &= \frac{1}{2}(-\omega_3 q_1 + \omega_1 q_3 + \omega_2 q_4) \\ \dot{q}_3 &= \frac{1}{2}(\omega_2 q_1 - \omega_1 q_2 + \omega_3 q_4) \\ \dot{q}_4 &= \frac{1}{2}(-\omega_1 q_1 - \omega_2 q_2 + \omega_3 q_3) \\ \dot{\omega}_1 &= \frac{B-C}{A}\omega_2\omega_3 + \frac{u_1}{A} \\ \dot{\omega}_2 &= \frac{C-A}{B}\omega_3\omega_1 + \frac{u_2}{B} \\ \dot{\omega}_3 &= \frac{A-B}{C}\omega_1\omega_2 + \frac{u_3}{C} \end{aligned} \quad (4.6)$$

where A, B, C are SimSAT II's mass MOI about the 1-axis, 2-axis, and 3-axis, respectively. The control vector \mathbf{u} is defined as

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (4.7)$$

where u_1, u_2 , and u_3 are the torques applied by SimSAT II's fans about the 1-axis, 2-axis, and 3-axis, respectively. Additionally, the effects of air drag on SimSAT II can also be accounted for in $\dot{\omega}_3$ by a fifth order polynomial function with the coefficients listed in

Table 4.1. These coefficients were estimated through the previously described empirical approach discussed in Section 3.4.3. To account for air drag in SimSAT II's dynamics, a polynomial function for air drag is added to the equation for $\dot{\omega}_3$ such that

$$\dot{\omega}_{3,new} = \dot{\omega}_3 + \dot{\omega}_{3,drag}. \quad (4.8)$$

The augmented cost functional for the fixed initial time, fixed initial state, fixed final state, free final time optimal control problem formulated in Eq. (4.1) through Eq. (4.7) can be written as

$$J' = t_f + \nu^T \psi(t_f, \mathbf{x}_f) - \int_{t_0}^{t_f} \lambda^T \dot{\mathbf{x}} dt, \quad (4.9)$$

where ν^T is an unknown constant multiplier vector and λ^T is an unknown Lagrange multiplier vector. The augmented cost functional is composed of the end point function G which is defined as

$$G = t_f + \nu_{q_1} q_{1,f} + \nu_{q_2} q_{2,f} + \nu_{q_3} (q_{3,f} - \sin \frac{\theta}{2}) + \nu_{q_4} (q_{4,f} - \cos \frac{\theta}{2}) \\ + \nu_{\omega_1} \omega_{1,f} + \nu_{\omega_2} \omega_{2,f} + \nu_{\omega_3} \omega_{3,f} \quad (4.10)$$

and the Hamiltonian H is defined as

$$H = \frac{\lambda_{q_1}}{2} (\omega_3 q_2 - \omega_2 q_3 + \omega_1 q_4) + \frac{\lambda_{q_2}}{2} (-\omega_3 q_1 + \omega_1 q_3 + \omega_2 q_4) \\ + \frac{\lambda_{q_3}}{2} (\omega_2 q_1 - \omega_1 q_2 + \omega_3 q_4) + \frac{\lambda_{q_4}}{2} (-\omega_1 q_1 - \omega_2 q_2 + \omega_3 q_3) \\ + \lambda_{\omega_1} \left(\frac{B-C}{A} \omega_2 \omega_3 + \frac{u_1}{A} \right) + \lambda_{\omega_2} \left(\frac{C-A}{B} \omega_3 \omega_1 + \frac{u_2}{B} \right) \\ + \lambda_{\omega_3} \left(\frac{A-B}{C} \omega_1 \omega_2 + \frac{u_3}{C} \right). \quad (4.11)$$

From Eq. (4.11) the state, costate, and stationary equations for this problem are calculated. The state equations are calculated by taking the first differential of the Hamiltonian with respect to λ^T and setting it equal to $\dot{\mathbf{x}}$. The state equations for this problem are defined in Eq. (4.5). The costate equations are calculated by taking the first differential of the Hamiltonian with respect to \mathbf{x} and setting it equal to $-\dot{\lambda}^T$. The

Table 4.1: Coefficients for SimSAT II's Air Drag Polynomial Function

Coefficient	Value
c_1	9.24×10^{-5}
c_2	5.73×10^{-20}
c_3	-2.33×10^{-3}
c_4	3.68×10^{-19}
c_5	-5.58×10^{-3}
c_6	0

costate equations for this problem are defined as

$$\begin{aligned}
 \dot{\lambda}_{q_1} &= \frac{1}{2}(\lambda_{q_2}\omega_3 - \lambda_{q_3}\omega_2 + \lambda_{q_4}\omega_1) \\
 \dot{\lambda}_{q_2} &= \frac{1}{2}(-\lambda_{q_1}\omega_3 + \lambda_{q_3}\omega_1 + \lambda_{q_4}\omega_2) \\
 \dot{\lambda}_{q_3} &= \frac{1}{2}(\lambda_{q_1}\omega_2 - \lambda_{q_2}\omega_1 + \lambda_{q_4}\omega_3) \\
 \dot{\lambda}_{q_4} &= \frac{1}{2}(-\lambda_{q_1}\omega_1 - \lambda_{q_2}\omega_2 - \lambda_{q_3}\omega_3) \\
 \dot{\lambda}_{\omega_1} &= \frac{1}{2}(-\lambda_{q_1}q_4 - \lambda_{q_2}q_3 + \lambda_{q_3}q_2 + \lambda_{q_4}q_1) + \lambda_{\omega_2}\frac{A-C}{B}\omega_3 + \lambda_{\omega_3}\frac{B-A}{C}\omega_2 \\
 \dot{\lambda}_{\omega_2} &= \frac{1}{2}(\lambda_{q_1}q_3 - \lambda_{q_2}q_4 - \lambda_{q_3}q_1 + \lambda_{q_4}q_2) + \lambda_{\omega_1}\frac{C-B}{A}\omega_3 + \lambda_{\omega_3}\frac{B-A}{C}\omega_1 \\
 \dot{\lambda}_{\omega_3} &= \frac{1}{2}(-\lambda_{q_1}q_2 + \lambda_{q_2}q_1 - \lambda_{q_3}q_4 + \lambda_{q_4}q_3) + \lambda_{\omega_1}\frac{C-B}{A}\omega_2 + \lambda_{\omega_2}\frac{A-C}{B}\omega_1.
 \end{aligned} \tag{4.12}$$

The stationary equations are calculated by taking the first differential of the Hamiltonian with respect to \mathbf{u} and setting it equal to zero. The stationary equations for this problem is defined as

$$\begin{aligned}
 0 &= \frac{\lambda_{\omega_1}}{A} \\
 0 &= \frac{\lambda_{\omega_2}}{B} \\
 0 &= \frac{\lambda_{\omega_3}}{C}.
 \end{aligned} \tag{4.13}$$

Boundary conditions for the costate equations are calculated by taking the first differential of the end point function with respect to the \mathbf{x}_f and setting it equal λ_f^T . The

boundary conditions for the costate equations are defined as

$$\begin{aligned}
 \lambda_{q_{1,f}} &= \nu_{q_1} \\
 \lambda_{q_{2,f}} &= \nu_{q_2} \\
 \lambda_{q_{3,f}} &= \nu_{q_3} \\
 \lambda_{q_{4,f}} &= \nu_{q_4} \\
 \lambda_{\omega_{1,f}} &= \nu_{\omega_1} \\
 \lambda_{\omega_{2,f}} &= \nu_{\omega_2} \\
 \lambda_{\omega_{3,f}} &= \nu_{\omega_3}.
 \end{aligned}
 \tag{4.14}$$

It is not possible to analytically solve these fourteen coupled highly nonlinear differential equations and the fourteen boundary conditions since SimSAT II is an asymmetric rigid body. For this research the PS-based optimal control solving tool, DIDO, is used to generate the optimal control solutions. In order to use DIDO to solve this optimal control problem, the problem must be reformulated into DIDO’s format as a set of MATLAB m-files [41]. Only the DIDO “main” m-file is changed for each test or experiment. The “Dynamics,” “Path,” “Events,” and “Cost” m-files are the same for all testing.

4.2 Open-Loop Optimal Control Implementation

SimSAT II implements an open-loop optimal control (OLOC) controller by using the Ground Station computer to run DIDO and calculate optimal control solutions prior to starting a rotation maneuver. The time and control inputs from DIDO’s solution are saved to the Ground Station’s hard disk drive in a .MAT file format. SimSAT II’s onboard MiniBox PC accesses the Ground Station PC over the wireless link and linearly interpolates the control solution from DIDO’s LGL point nodal distribution into 0.2-second spaced discretization that matches the cycle time of SimSAT II’s Simulink optimal control model. The MiniBox PC then provides the control solutions sampled at 0.2 seconds to the Simulink control model running on the dSpace MicroAutoBox using a MATLAB script that calls the “mlib” function library. Since the control inputs are updated once every 0.2 seconds, the control input is described as being a “zero-order

hold” approximation of the linearly interpolated DIDO solution. This process is shown graphically in Figure 4.1.

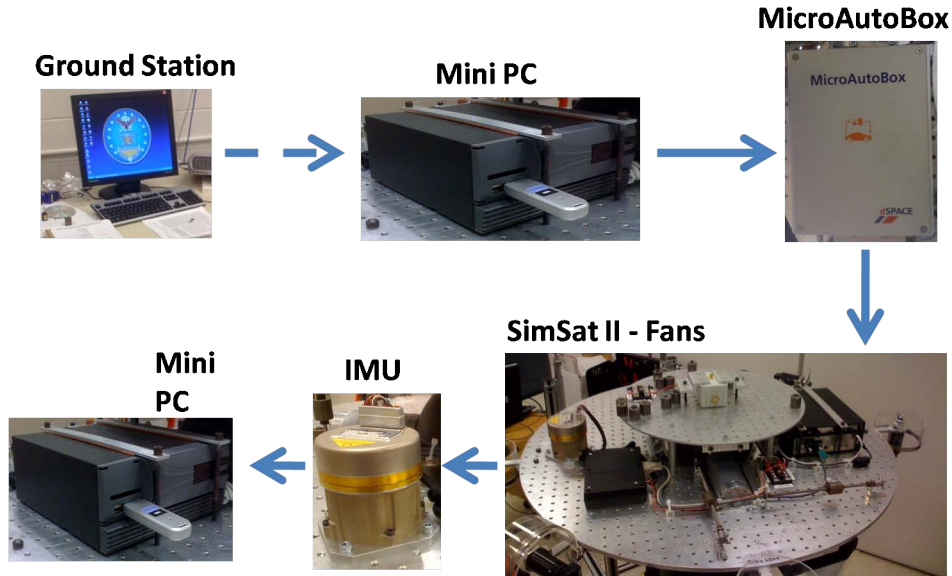


Figure 4.1: Implementation of OLOC Controller on SimSAT II

As mentioned in Section 3.2, the ControlDesk program running on the MiniBox PC provides users with a GUI to input Simulink control models into the dSpace MicroAutoBox. ControlDesk allows users to create customizable GUIs in which the user can record any Simulink model variable. For this research, the ControlDesk GUI records the time history of all angular positions, angular rates, and control inputs. Additionally, the dSpace MicroAutoBox runs Simulink control models using an internal clock that is separate from the timing used in DIDO. In order to rectify the difference between the DIDO and dSpace MicroAutoBox clocks, the dSpace MicroAutoBox internal clock time is used to normalize the states recorded by ControlDesk.

4.3 Real-Time Optimal Control Implementation

RTOC functions similar to traditional closed-loop controller in that it uses current state information to determine a difference between the current and desired states from which the controls are computed. Instead of calculating a single set of control values for that moment in time, the RTOC controller calculates the entire open-loop control and state history. This control history includes the current and all future controls necessary

to achieve the desired states. Therefore, a RTOC controller can be thought of in terms of two control loops, where the “outer” open-loop controller is designed to optimally achieve a final set of desired states and the “inner” closed-loop controller is designed to maintain movement along the most recently calculated optimal path, as shown in Figure 4.2. As the “outer” open-loop controller progresses through the optimal control solution, the current states and controls will be sampled by the “inner” closed-loop controller. If a norm of the difference between the sampled states and optimal trajectory is greater than some threshold value, then the “inner” closed loop controller will recompute a new optimal control solution. Otherwise, the system will maintain its current optimal control solution.

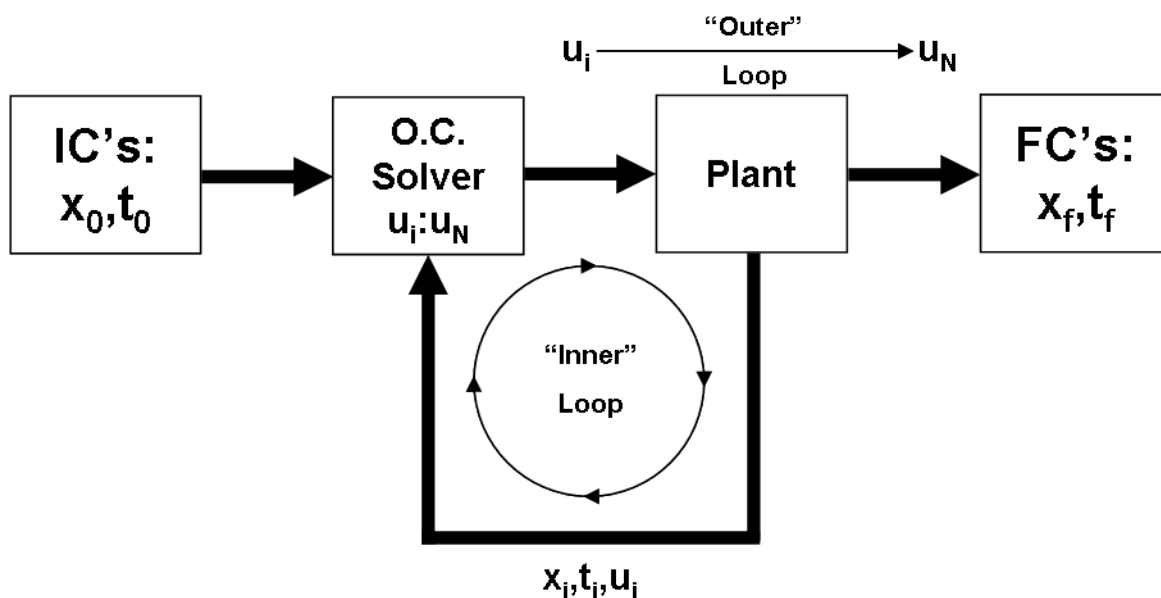


Figure 4.2: RTOC Control Loops

On SimSAT II, the “outer” open-loop controller is implemented using the traditional optimal controller process as described in Section 4.2, while the “inner” RTOC control loop uses a modified version of this process. Instead of just using DIDO to only calculate the optimal control solution prior to the starting the rotation maneuver, the “inner” RTOC control loop recalculates the optimal control solution during the maneuver. The OLOC MATLAB script calling the “mlib” function library to input control solutions is modified to use the “mlib” function library to also read SimSAT II’s current angular position and rates prior to inputting the first node a control solution. The

MiniBox PC uses the wireless link to save the recorded state information to the hard disk drive of the Ground Station PC. The Ground Station PC then loads the new state information into the DIDO problem files and recomputes a new solution that is saved to the hard disk drive. This process is shown graphically in Figure 4.3.

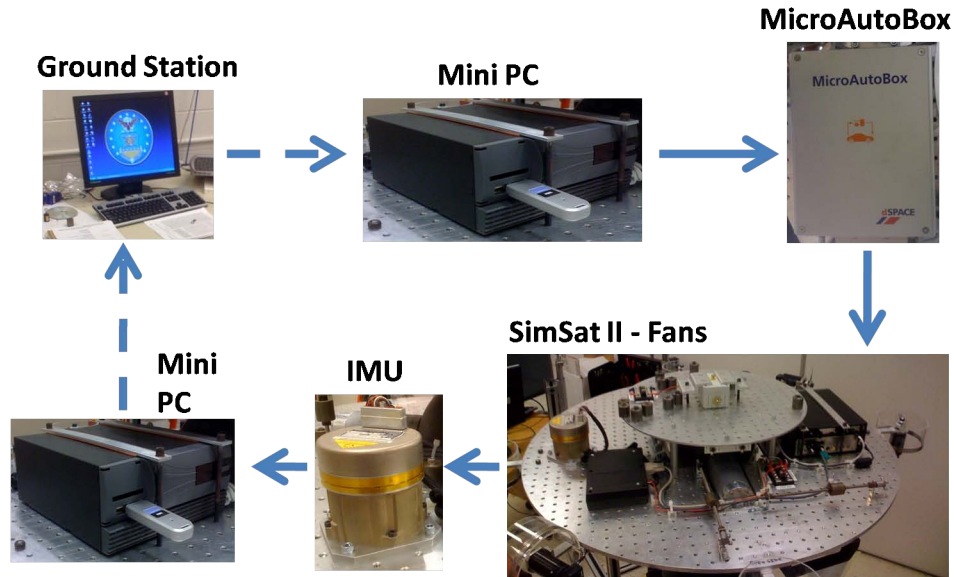


Figure 4.3: Implementation of RTOC Control on SimSAT II

Depending on the complexity of the optimal control problem, DIDO can take anywhere from one second to several minutes to compute a new optimal control solution. This computational time delay requires one to shift forward in time from when the new “inner” loop optimal control solution was calculated to when the solution is actually implemented. The time delay is accounted for in the control solution by truncating the beginning of the control solution for the computational time duration. A side-effect of truncating the new control solution is that it introduces a discontinuity in the control input, as seen in Figure 4.4, where the black circles represent when a new control iteration takes place as calculated by DIDO solution nodes and the clear circles represent when a new control iterations take place as implemented by SimSAT II.

The “inner” and “outer” control loops in the RTOC controller also use logical switches to make decisions how to use new optimal control solutions and when to terminate a maneuver. As shown in Figure 4.5, the control logic for the “inner” closed-loop decides whether or not to use a newly computed optimal solution based on an error

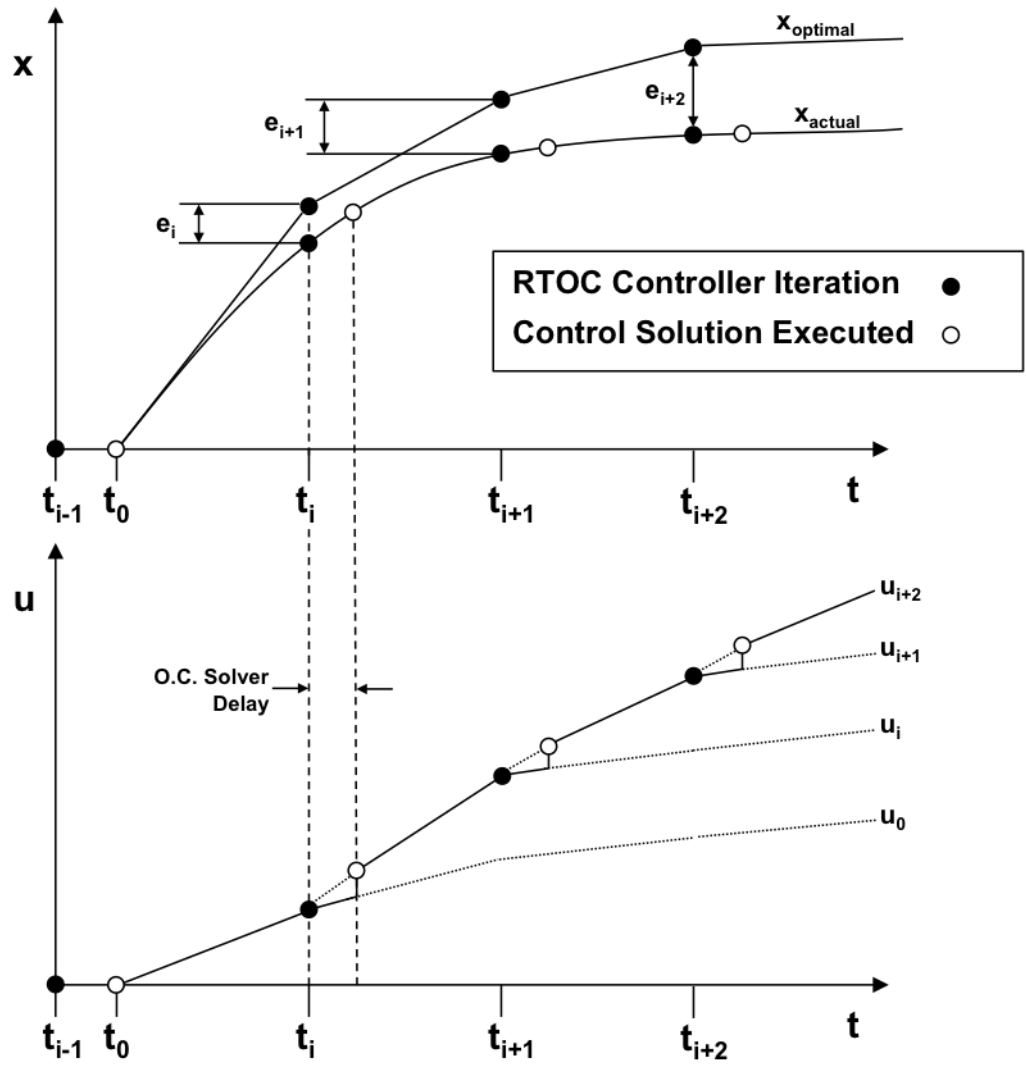


Figure 4.4: Implementation of RTOC Control on SimSAT II

metric ϵ_{inner} derived from the estimated state history from the previous optimal control solution and the current state measured. The “inner” control loop error metric is defined as

$$\epsilon_{inner} = a(1 - q_{4,e}) + b\|\omega_e\|, \quad (4.15)$$

where $\|(-)\|$ is the L2 norm of $(-)$, ω_e is the difference between the estimated angular rate from the previous optimal control solution and the measured angular rate, $q_{4,e}$ is the fourth value in the error quaternion between the estimated quaternion parameter set from the previous optimal control solution and the measured quaternion parameter set, and a and b are weighting scalars both with the value of 50. When ϵ_{inner} was less than the threshold value of 0.1, new optimal control solutions are ignored. This logic reduces large magnitude discontinuities between new optimal control solutions and previous control solutions, such as those seen in Figure 4.4. The values of the weighting scalars and threshold values were determined through trial and error by modeling a RTOC controller performing a 180° yaw axis rotation. The weighting scalars and thresholds were manipulated until the final state error in $q_{4,e}$ was less than 0.005 and the final state error in each ω_i was less than 0.001 rad/s. Approaches to determine a methodology in setting these values is suggested as future work.

The control logic for the “outer” open-loop decides whether or not to stop computing new optimal control solutions based on an error metric ϵ_{outer} derived from the desired final states and the received actual state information used for each optimal control iteration. This research also uses Eq. (4.15) for the “outer” loop error metric. However, for the “outer” loop error metric, ω_e is the difference between the desired final angular rate and the measured angular rate, $q_{4,e}$ is the fourth value in the error quaternion between the desired final quaternion parameter set and the measured quaternion parameter set, and a and b are weighting scalars with the value of 1 and 0 respectively. When ϵ_{outer} was less than the threshold value of 0.005, the RTOC control stops recalculating optimal control solutions and completes the optimal control maneuver with its current control solution. The purpose of this logic is to signify that SimSAT II has completed the rotational maneuver and achieved the desired final states. Theoretically, the RTOC controller could

also be used to maintain steady-state about the desired set of final states, but this was not investigated.

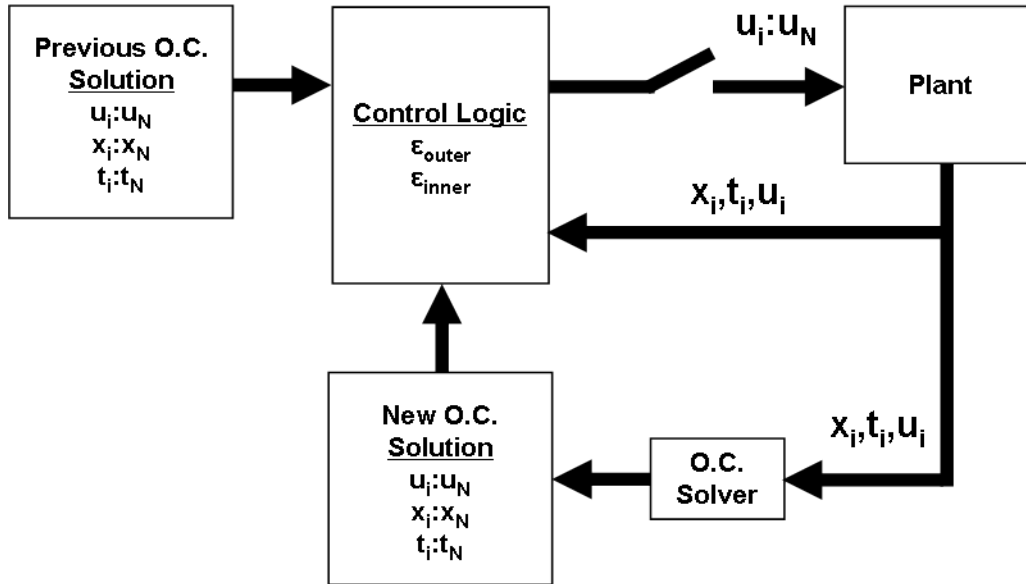


Figure 4.5: RTOC Controller Logic

4.4 *SimSAT II Modeling and Simulation*

A simulation model of the traditional optimal control and RTOC controllers for SimSAT II was created in MATLAB. ODE45 is used to propagate both open and closed loop optimal control solutions over time. The simulated system use the same EOMs as used by DIDO. Since the OLOC and RTOC controllers share the same “outer” open-loop control function, both controllers can be tested from the same model because the initial RTOC optimal control solution is also the traditional optimal control solution as well. Path error is not directly introduced into the simulation model, but is instead introduced through using ODE45 to propagate a DIDO calculated optimal control solution. More accurate optimal control solutions, calculated by DIDO with more LGL nodes, will experience less path error when the solution is propagated by ODE45 than when less accurate optimal control solutions, calculated by DIDO with fewer LGL nodes, are used.

V. Results

THIS chapter describes and analyzes the simulation and experimental results for two open-loop optimal controllers and a RTOC controller. The first section of this chapter describes the simulation and experimental testing procedure and provides guidance for interpreting figures. The next section presents the performance results of SimSAT II's open-loop optimal controller for a set of simulation and experimental reorientation maneuvers. These results will then be compared with the results of SimSAT II's eigenaxis rotation optimal controller for the same reorientation maneuvers. The performance results of SimSAT II's RTOC controller are presented next and compared with the results of both a open-loop optimal control controller and the eigenaxis controller for the same maneuvers. The final section presents the performance results of SimSAT II's RTOC controller as model and problem formulation parameters are varied.

5.1 Test Procedure

This research performs simulation and experimental testing for fixed initial time, fixed initial state, fixed final state, free final time optimal control problems on the SimSAT II simulated satellite. All optimal control problems tested in this research are described by the Roll-Pitch-Yaw Euler Angle rotation between the fixed initial states described in Eq. (4.3) and the fixed final states described in Eq. (4.4). For this research, only 45°, 90°, 180° yaw axis rotations are tested as shown in Figure 5.1. Each optimal control problem tested uses the performance index shown in Eq. (4.1) which minimizes the time taken to perform the desired yaw axis rotation.

Both experimental and simulation results are collected using either an open-loop optimal control (OLOC), an eigenaxis rotation optimal control (EROC), or a RTOC controller. The OLOC controller, described in Section 4.2, is effectively only the “outer” loop of a RTOC controller. The OLOC controller calculates and executes a single optimal control solution that was calculated using DIDO prior to starting the reorientation maneuver. The EROC controller is functionally the same as an OLOC controller, and only differs in the formulation of the optimal control problem being solved. An EROC controller only allows rotation about the eigenaxis formed by each optimal control problem's

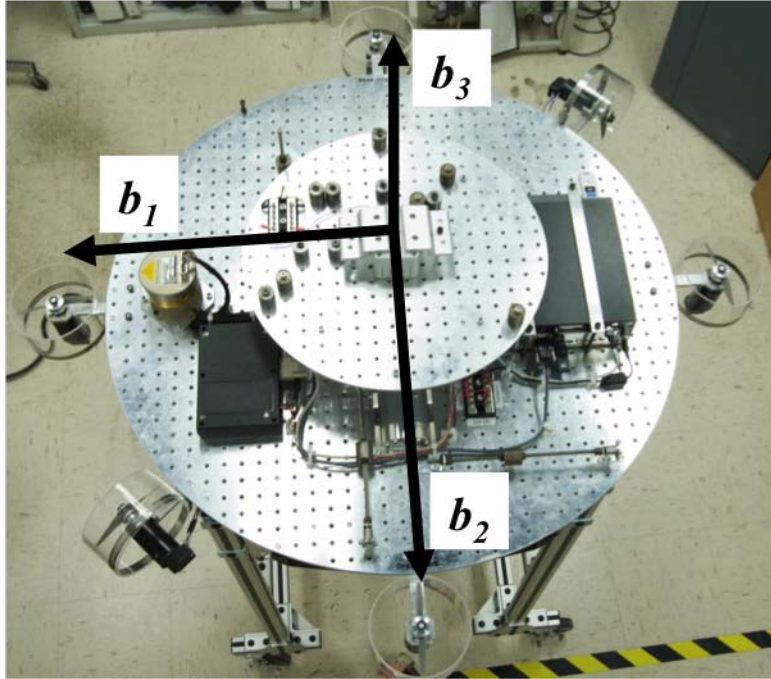


Figure 5.1: SimSAT II Body-Frame Axes

desired yaw axis rotation. The EROC controller will find the minimum time required to rotate about the smallest angular path, since an eigenaxis rotation uses the smallest angular path between two orientations. An eigenaxis rotation is the most widely used maneuver to accomplish the reorientation of a spacecraft, therefore the EROC controller's performance results will serve as a baseline for comparison to the OLOC and RTOC controller results. The RTOC controller, described in Section 4.3, continuously uses DIDO to solve the optimal control problem during the course of the reorientation maneuver based on SimSAT II's current state values.

Simulation and experimental testing are performed for SimSAT II's OLOC and EROC controllers, while the RTOC controller performs in simulation testing only because a working RTOC controller was not able to be implemented on SimSAT II within the timeline of this research. Results presented in this chapter will be specifically labeled as whether they were produced by simulation or experimental testing. SimSAT II simulation testing is conducted by linearly interpolating a discrete optimal control solution calculated by DIDO and using the MATLAB ODE45 function to propagate the state solution over time. The EOM used to simulate SimSAT II are identical to the

EOM used by DIDO to calculate an optimal control solution. Experimental testing is conducted using SimSAT II simulated satellite and the methods described in Section 4.2. The LN-200 gyroscope measures angular rotation rate information during experimental testing, which is post-processed to calculate SimSAT II's quaternion parameters.

The RTOC controller is primarily characterized by its solution update time, referring to the simulation time that takes place between each iteration of solving the optimal control problem. For example, if a RTOC controller has a 1-second update time, then in simulation time it theoretically takes DIDO one second to solve the optimal control problem, even though it may take quite a bit longer or even shorter. If the RTOC controller uses a “real-time” update time, then the simulation time between iterations is based on DIDO's actual computation time. Generally, it takes DIDO anywhere from one second to several minutes to calculate an optimal control solution, however, assuming DIDO takes one second to calculate a solution is not necessarily a bad assumption since DIDO is currently running in a MATLAB environment as a single process on a desktop computer. If DIDO were to be used on-board an actual spacecraft, a parallelized version would be rewritten using a compiled programming language and would significantly reduce DIDO's computation time.

For all figures displaying the state and control history results of the OLOC and EROC controllers, colored circles represent the discrete states and control histories as calculated by DIDO using a LGL nodal distribution. For simulated results of the OLOC and EROC controllers, dash-dot lines represent the state and control histories that were propagated using the MATLAB ODE45 function. For experimental results of the OLOC and EROC controllers, dash-dot lines represent the states and controls as measured by the LN-200 gyroscope on-board SimSAT II.

For all figures displaying the state and control history results of RTOC controllers, colored circles represent the discrete states and control histories calculated by DIDO using a LGL nodal distribution. Dash-dot lines in RTOC figures represent the state and control histories that were propagated by the MATLAB ODE45 function using the initial optimal control solution calculated by DIDO. The initial optimal control solution iteration for the RTOC controller is the same optimal control solution that is used for the

OLOC controller. Solid lines in RTOC figures represents the states and controls propagated by the MATLAB ODE45 function and formed by concatenating each iteration of DIDO solving the optimal control problem.

In some figures, a three-dimensional (3-D) animation of the maneuver being performed will also be shown to provide a better visual interpretation of the simulated or measured quaternion states. Each animation contains a red and blue 3-D shape representing SimSAT II, and cyan, green, and magenta dotted lines representing the path of SimSAT II's roll, pitch, and yaw body frame axes, respectively, during the course of the maneuver. Each animation also contains a black circle representing where SimSAT II's body frame axes were pointing at the start of the maneuver, and a dark green circle representing where SimSAT II's body frame axes are pointed at the end of the maneuver. The unit-less x , y , and z axes displayed in each animation figure represents SimSAT II's inertial frame.

5.2 Open-Loop Optimal Control

Simulation and experimental results were collected for SimSAT II's OLOC controller performing 45°, 90°, and 180° yaw axis rotation maneuvers. The optimal control path of each maneuver was calculated in DIDO prior to the maneuver using the parameters listed in Table 5.1. Constraints for the maximum control torque were set to small values in order to reduce control discontinuities. Angular rate limits were set to small values to reduce the effects of air drag. Using the 5th order polynomial coefficients listed in Table 4.1, a maximum angular rate of 0.1 rad/s will result in a maximum torque caused by air drag equivalent to $<6 \times 10^{-7}$ N-m. Even though the effects of drag are extremely small, they are still accounted for when calculating an optimal control solution. Preliminary testing had demonstrated that the desired optimal control path for the 90 and 180 degree rotational maneuvers would violate SimSAT II's rotational constraints due to the air bearing. Therefore, quaternion parameters q_1 and q_2 , which approximately correspond to rotation about the roll and pitch Euler angles, respectively, are subject to a maximum quaternion magnitude of ± 0.1 , which is approximately corresponds to a rotation of ± 11.5 degrees about the Roll or Pitch Euler Angles.

Table 5.1: OLOC Simulation and Experimental Testing Parameters

Parameter	Yaw Axis Rotation		
	45°	90°	180°
DIDO nodes	50	50	50
Maximum Control Torque (N-m)	± 0.025	± 0.025	± 0.025
Maximum Angular Rate (rad/s)	± 0.1	± 0.1	± 0.1
Maximum Quaternion q_3, q_4	± 1	± 1	± 1
Maximum Quaternion q_1, q_2	± 1	± 0.1	± 0.1

5.2.1 Open-Loop Optimal Control Results. Figure 5.2 shows the simulated results for the quaternion values and angular rotation rates of the SimSAT II model during a OLOC controller for a 45° yaw axis rotation. The simulated states resulting from the propagation of the DIDO solution match very closely with DIDO’s estimated solution for the states. There is still a small noticeable difference between the DIDO estimated solution and the modeled propagated solution in Figure 5.2, where the colored lines representing the modeled propagated solution does not go exactly through the center of the colored circles representing the DIDO estimated solution. This occurs because DIDO uses Lagrange interpolating functions to estimate the optimal control solution based on a user defined amount of LGL nodes as discussed in Section 2.4.2. The DIDO solution shown in Figure 5.2 uses 50 LGL nodes. For a larger number of LGL nodes the DIDO solution will be in closer agreement with the simulated results because the optimal control solution will more closely resemble a “bang-bang” controller. Bilimoria and Wie have ascertained that the results for the optimal control solution to reorientation of an axisymmetric spacecraft will always appear “bang-bang” [8]. As the optimal control solution appears to be less “bang-bang” like, the less optimal the solutions become. Therefore, if the number of LGL nodes used by DIDO were decreased, it is expected that the DIDO solution will differ more from the simulated results because the optimal control solution will appear less “bang-bang” like. This effect will be discussed in detail in Section 5.4.

Figure 5.3 shows the experimental results for the quaternion values and angular rotation rates measured by SimSAT II during a OLOC controller for a 45° yaw axis rotation. The first thing that is apparent from Figure 5.3 is that the OLOC controller

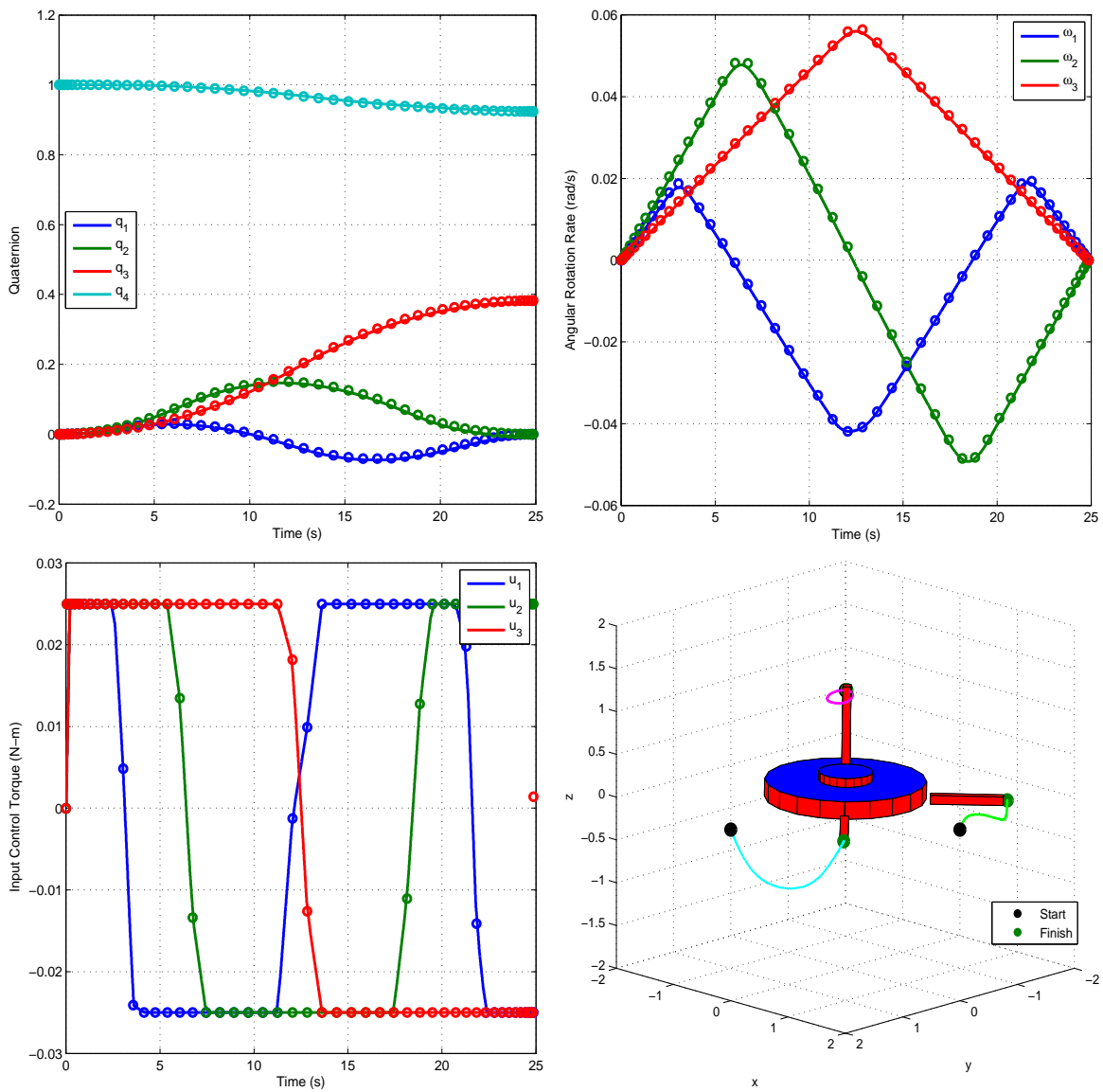


Figure 5.2: SimSAT II Simulation Results - OLOC Controller 45° Maneuver

solution is different than that shown in Figure 5.2. As a result of a different controller solution, the quaternion and angular rotation rate state history for the experimental testing shown in Figure 5.3 is distinctly different from the state history of the simulation testing shown in Figure 5.2. This occurred because DIDO found two different but valid solutions to the same optimal control problem. Pseudospectral-based optimal control solvers such as DIDO are sensitive to parameter changes, whether it be parameter changes in the formulation of the problem or changes to the parameters within the solvers' internal formulation. As a result, the solutions found by DIDO can infrequently be volatile and subject to change from one call of the solver to another. Suggested methods for minimizing this effect is discussed in Section 5.4. The optimal controller currently implemented on SimSAT II requires that DIDO calculate the solution to the maneuver prior to implementing it. In other words, you cannot take a previously recorded DIDO solution and implement it, instead a new DIDO solution is computed each time. This implementation works well for RTOC, since the DIDO is continuously solving the optimal solutions, however the resulting inconsistency in DIDO solutions has drawbacks for traditional optimal control.

Another significant difference between the SimSAT II's simulation and experimental testing is the method used to input the control to the system. The controls used by SimSAT II for simulation testing shown in Figure 5.2 uses linear interpolation of the controls between DIDO's LGL nodes and updates the control value every 0.005 seconds. The controls used by SimSAT II for experimental testing shown in Figure 5.3 uses linear interpolation of the controls between the DIDO LGL nodes, but the controls are subject to a zero-order hold discontinuity every 0.2 seconds. The physical limitations of SimSAT II's hardware, as discussed in Section 3.1.2, prevent the controls from consistently being externally updated faster than once every 0.2 seconds. The optimal control solution calculated by DIDO can be thought of as a discrete solution that is linearly interpolated between each node. By implementing the control solution with a 0.2 second zero-order hold, this effectively changes the optimal control solution and introduces error.

The zero-order hold approach for implementing the control solution is just one of the reasons why the experimental results do not closely match DIDO's estimated solu-

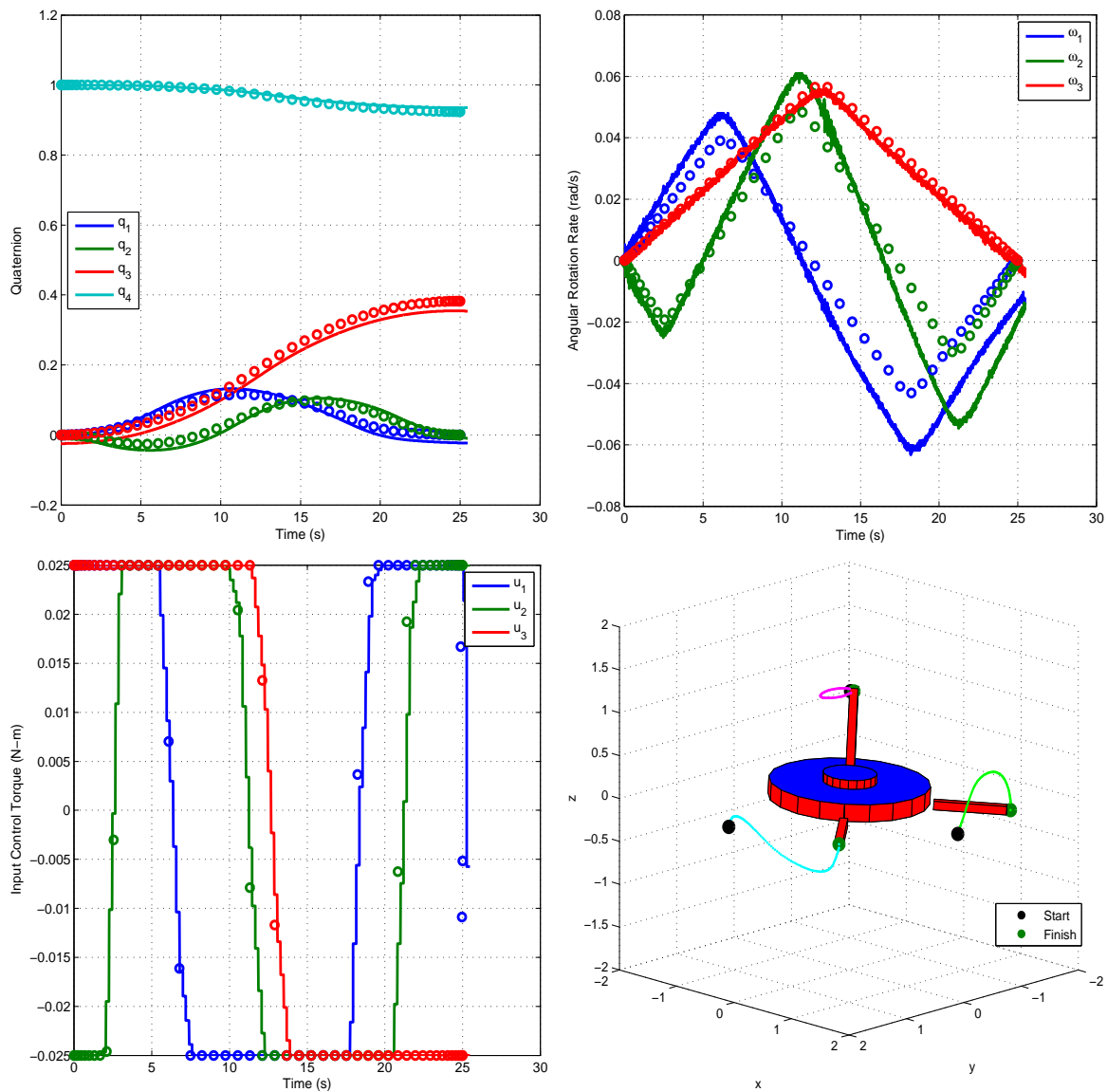


Figure 5.3: SimSAT II Experimental Results - OLOC Controller 45° Maneuver

tion. There are several other error causing factors that also need to be accounted for. The angular rotation rates shown in Figure 5.3 show that the yaw axis rotation shows the least error, while the roll and pitch axes have significantly more error. As discussed in Section 3.4.3, thruster characterization at low fan RPM rates are inconsistent due to small Re of the fans' operating environment, therefore the thrust produced by the fans during experimental testing is likely to be inaccurate. There is additional error in the thrust characterization for the roll and pitch axis fans because the yaw axis thruster characterization values were used for all of the fans. The roll and pitch axis fans should be characterized separately to reduce this error. Similarly, torque due to air drag was only accounted for about the yaw axis, therefore torque due to air drag should also be characterized for the roll and pitch axes and included into the system's EOMs. Uncertainty in the estimation of SimSAT II's MOI matrix is also a likely source of error. The effects of MOI uncertainty will be discussed in Section 5.4.

OLOC simulation and experimental testing results for 90° and 180° yaw axis rotation maneuvers are shown in Figures 5.4, 5.5, 5.6, and 5.7. The results of these maneuvers are characterized by the error sources highlighted for the OLOC controller's 45° yaw axis rotation maneuver shown in Figure 5.2 and Figure 5.3. The OLOC simulation testing performance indices (PI) for all three yaw axis rotations is listed in Table 5.2. The units of the PIs are in seconds because, the PIs are only a function of the maneuver's final time.

Table 5.2: Simulation Results - OLOC Performance Indices

Yaw Axis Rotation	OLOC PI (s)
45°	25.00
90°	35.41
180°	50.92

5.2.2 Eigenaxis Rotation Optimal Control Results. Both simulations and experimental tests were performed using SimSAT II's EROC controller for 45° , 90° , and 180° yaw axis rotation maneuvers. The only difference between the OLOC controller and the EROC controller is that in the DIDO formulation of optimal control problem. For the EROC controller, the maximum angular rotation rate about the Roll and Pitch

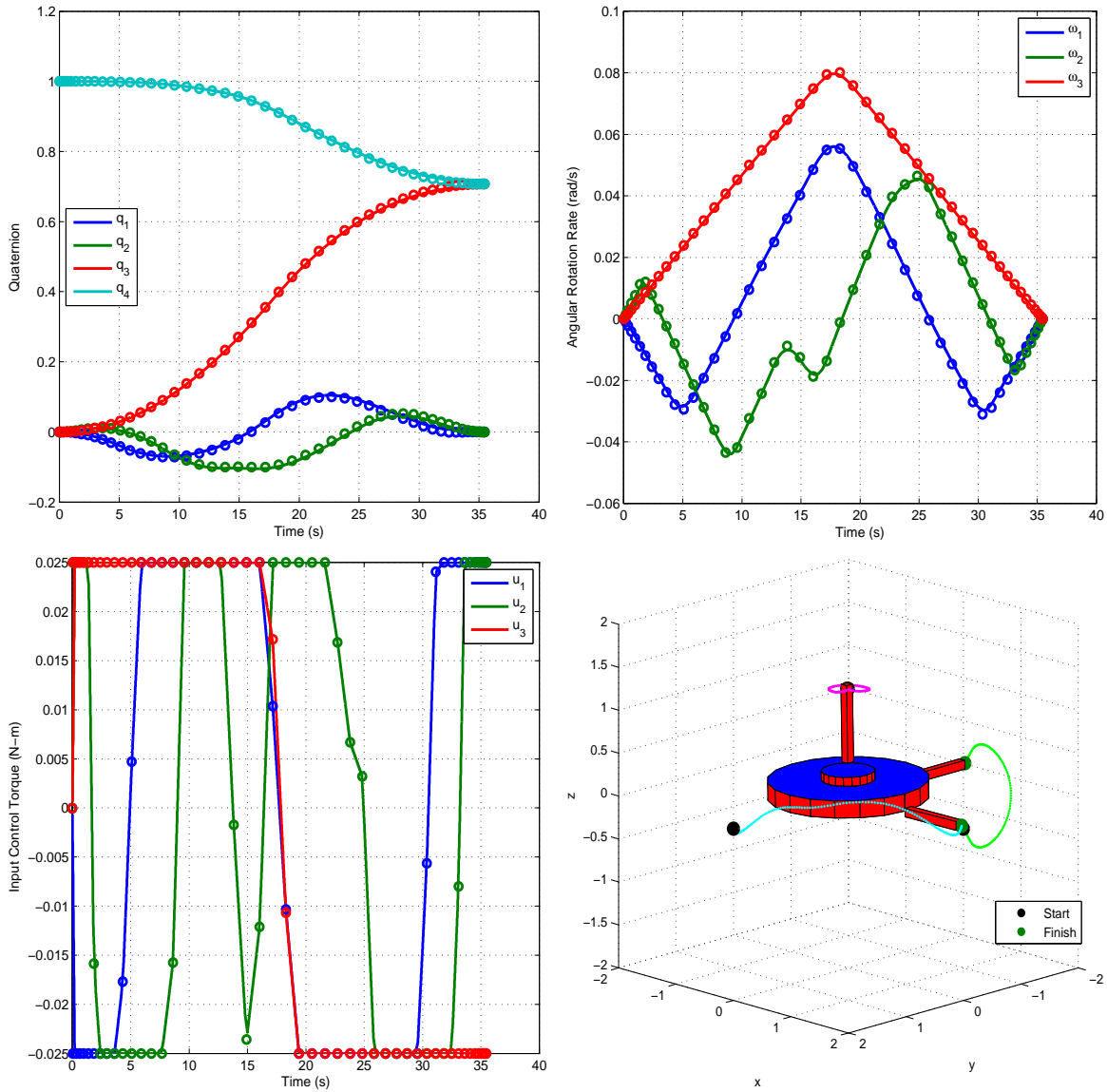


Figure 5.4: SimSAT II Simulation Results - OLOC Controller 90° Maneuver

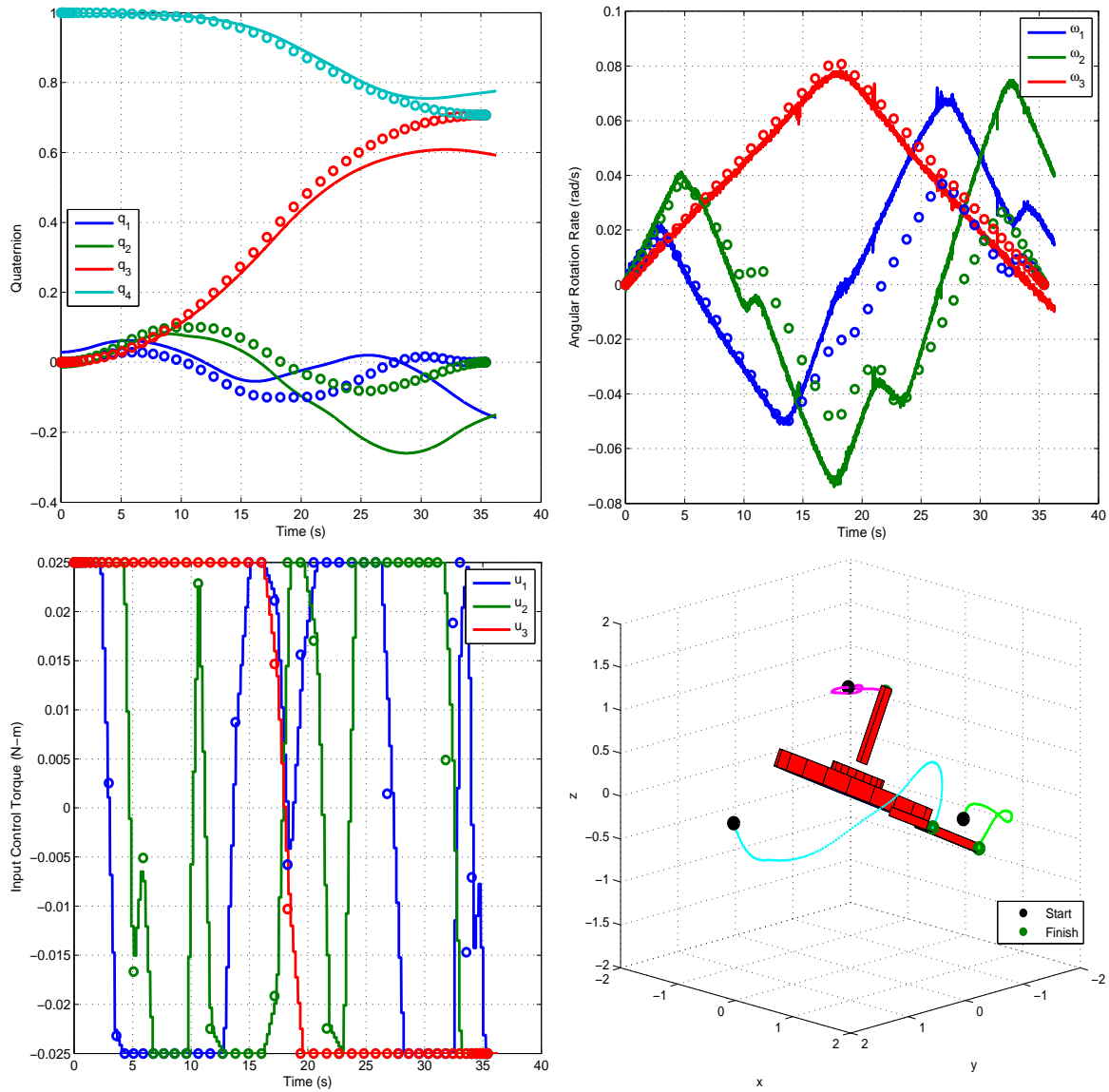


Figure 5.5: SimSAT II Experimental Results - OLOC Controller 90° Maneuver

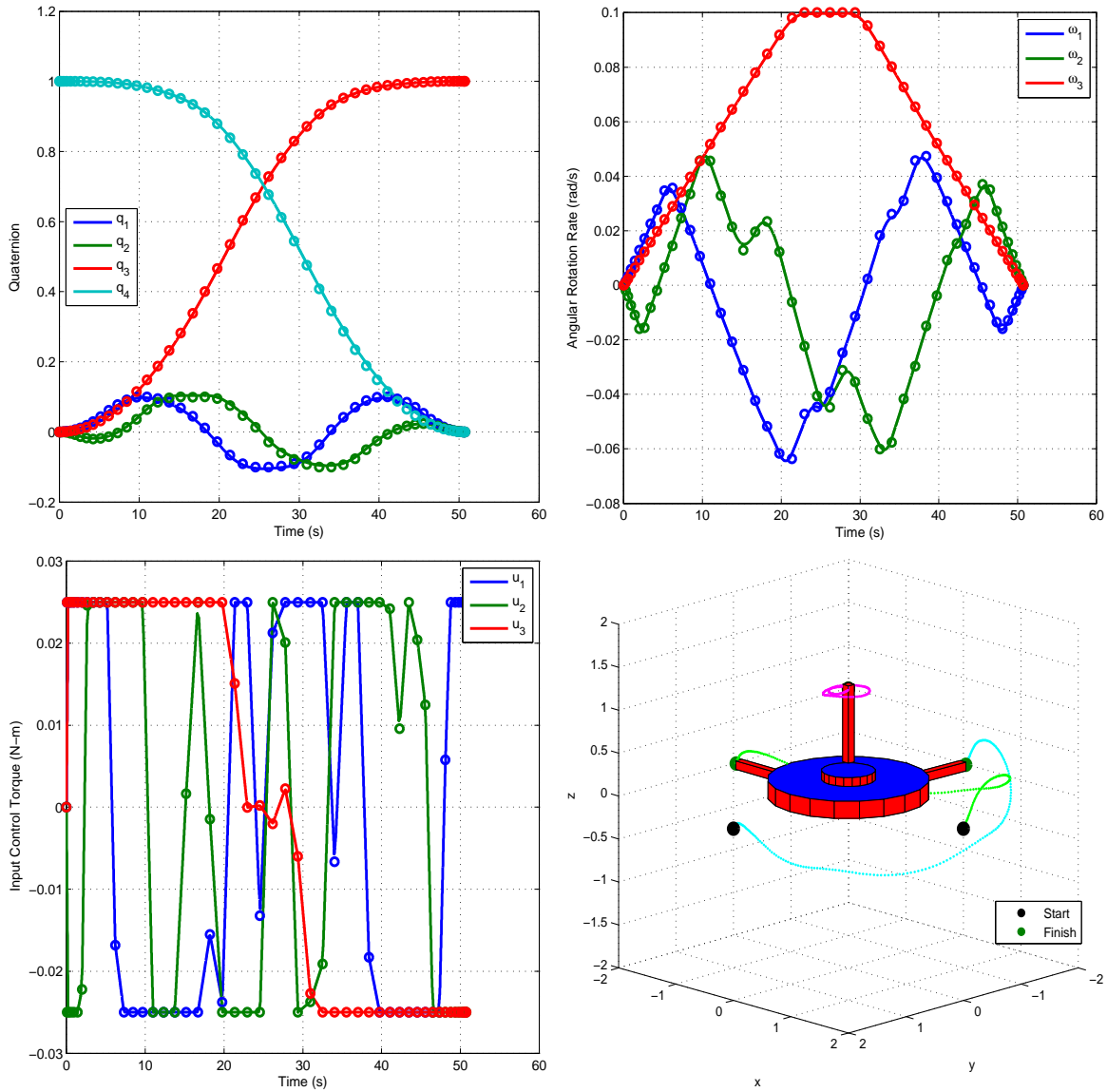


Figure 5.6: SimSAT II Simulation Results - OLOC Controller 180° Maneuver

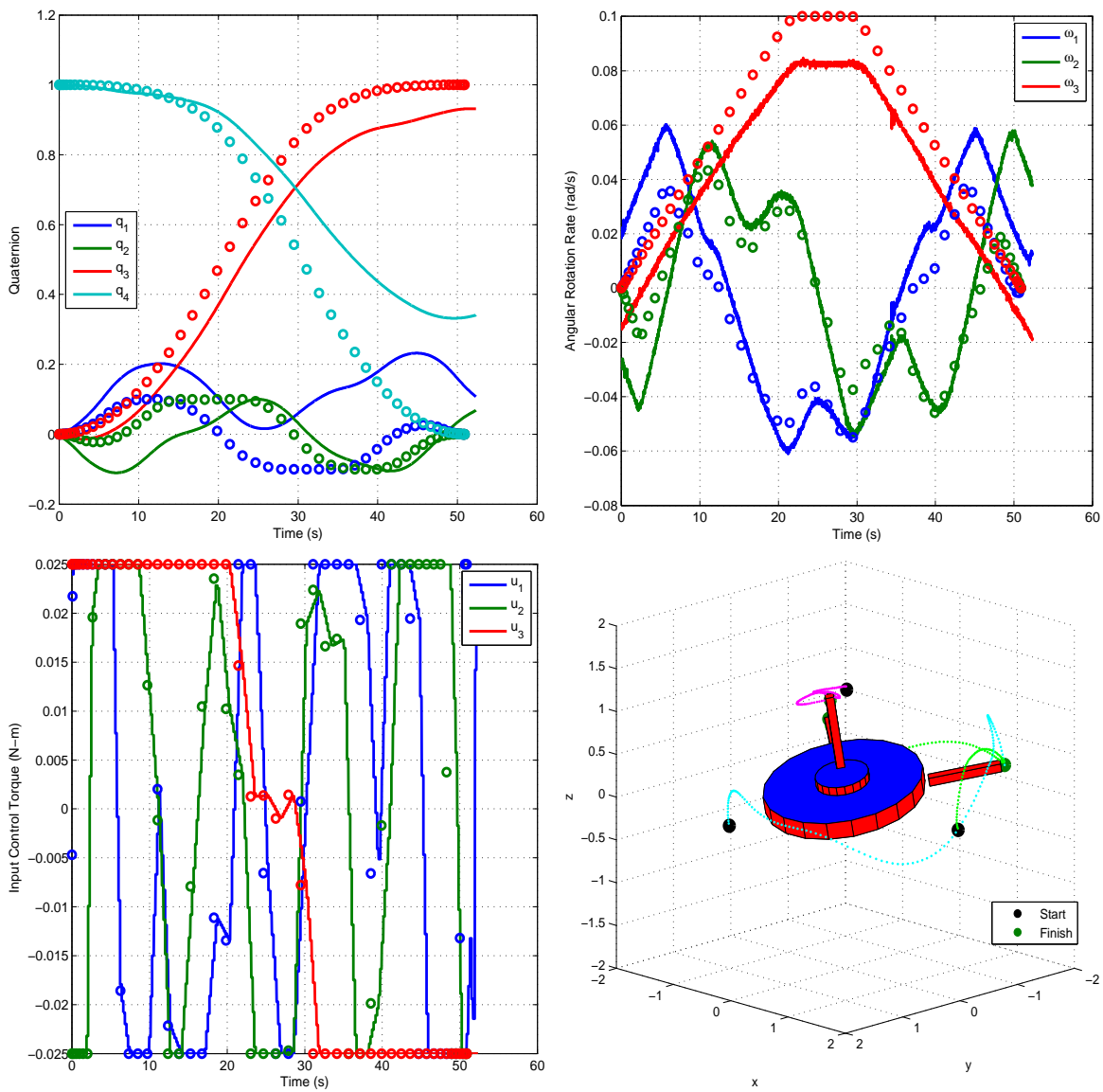


Figure 5.7: SimSAT II Experimental Results - OLOC Controller 180° Maneuver

axes are restricted to 0 rad/s. The yaw axis is the eigenaxis for 45°, 90°, and 180° yaw axis rotation maneuvers as described in Section 5.1.

The simulation testing results of SimSAT II’s EROC controller performing a 45° yaw axis rotation maneuver is shown in Figure 5.8. Similar to the OLOC controller’s simulation results in Section 5.2.1, the EROC controller’s simulation results shown in Figure 5.8 closely matches the DIDO estimate for the states. Additionally, Figure 5.8 shows that the minimum time control input for eigenaxis rotation maneuver maneuvers closely resembles a “bang-bang” control solution.

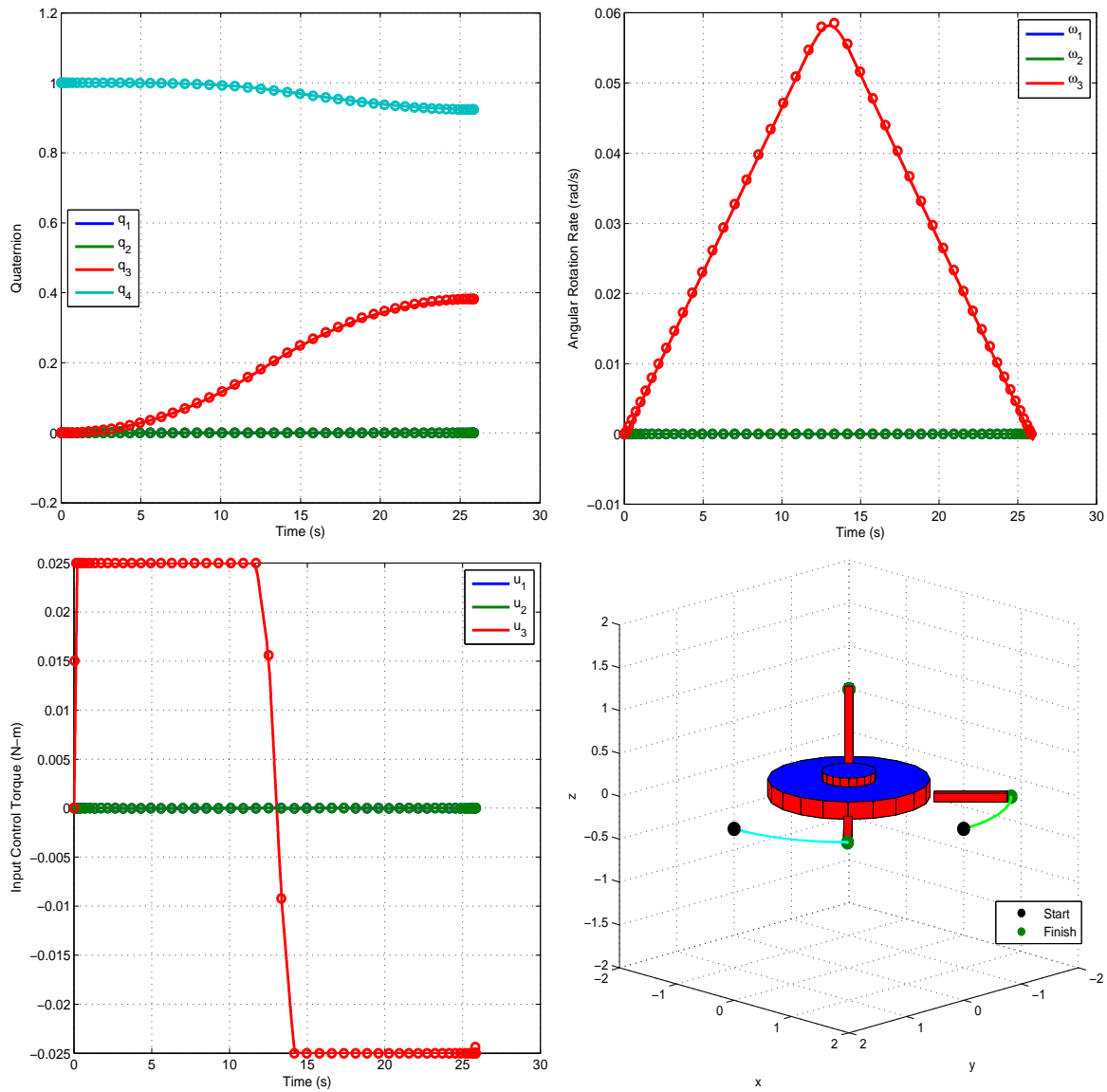


Figure 5.8: SimSAT II Simulation Results - EROC Controller 45° Maneuver

The experimental testing results of SimSAT II's EROC controller performing a 45° yaw axis rotation maneuver, shown in Figure 5.9, highlight errors in the SimSAT II system that is not accounted for in the system's EOM. Unlike the experimental results of the OLOC controller in Section 5.2.1, there is significant error in the yaw axis thruster, suggesting that the yaw axis thruster characterization needs to be further refined. The small angular rotation rates about the Roll and Pitch suggest that SimSAT II may have large enough POI terms in its MOI matrix that they cannot be ignored in the system's EOM, as done in Eq. (4.5) and (4.6).

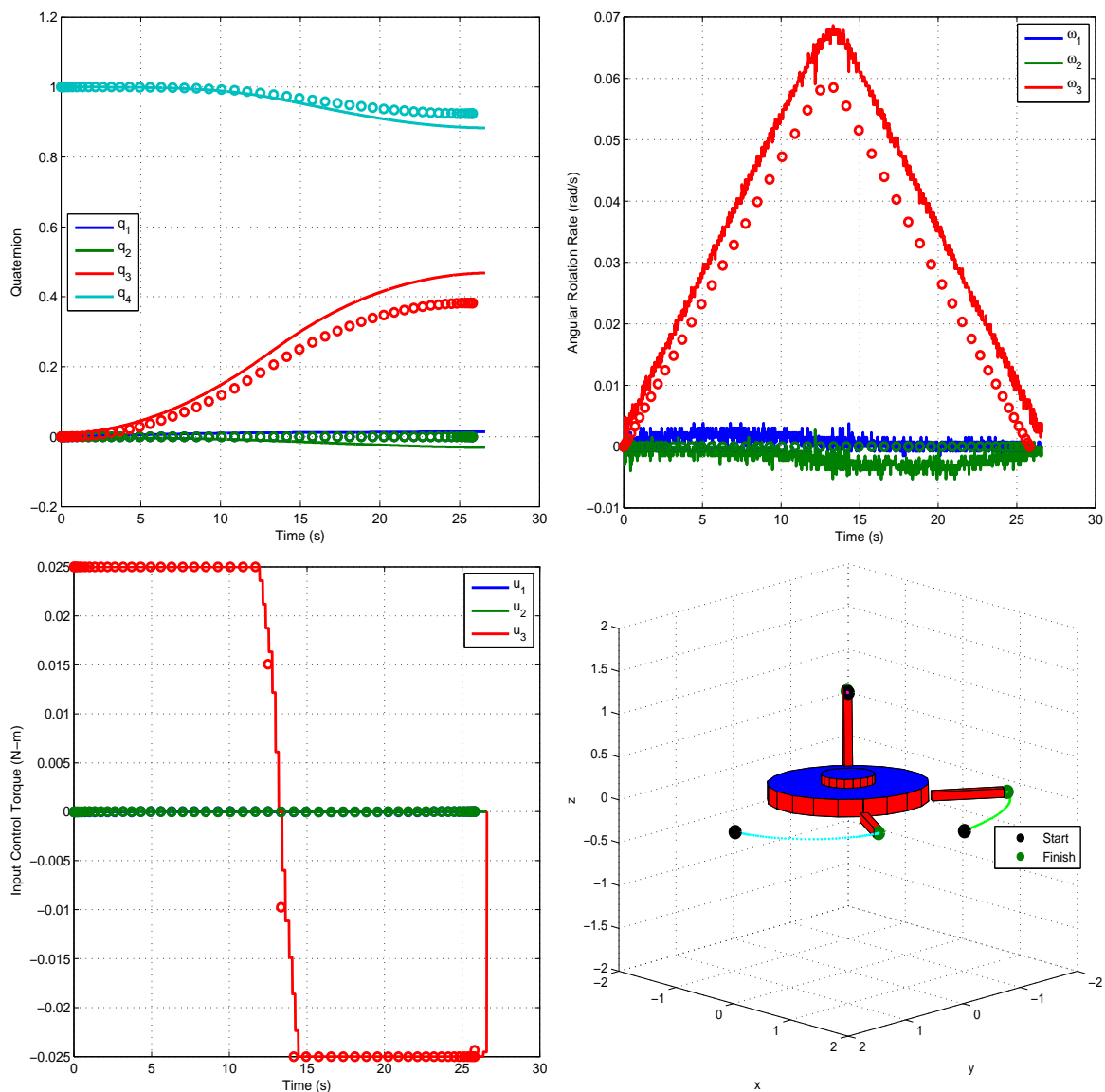


Figure 5.9: SimSAT II Experimental Results - EROC Controller 45° Maneuver

EROC simulation and experimental testing results for 90° and 180° yaw axis rotation maneuvers are shown in Figures 5.10, 5.11, 5.12, and 5.13. The appearance of these results are characterized by the same error sources highlighted for the EROC controller’s 45° yaw axis rotation maneuver shown in Figure 5.8 and 5.9. The resulting performance indices of the OLOC and EROC controllers for all three yaw axis rotations are listed in Table 5.3. Across the board, the rotation maneuver produced by the OLOC controller created an approximately 3% increase in performance over a comparable eigenaxis maneuver produced by the EROC controller.

Table 5.3: OLOC and EROC Controller Simulation Performance Index Comparison

Yaw Axis Rotation	EROC Final Time (s)	OLOC Final Time (s)	Δt
45°	25.83	25.00	-3.21%
90°	36.53	35.41	-3.07%
180°	52.65	50.92	-3.29%

5.3 Real-Time Optimal Control Results

SimSAT II’s RTOC controller is simulated with real-time control solution updates performing the same 45°, 90°, and 180° yaw axis rotation maneuver as performed by the OLOC and EROC controllers. For a RTOC controller with real-time control solution updates, DIDO uses the parameters listed in Table 5.4 to continuously recalculate the optimal control and updates the optimal control solution being propagated by ODE45 as soon as the previous control solution iteration was calculated.

Table 5.4: RTOC Controller Simulation Testing Parameters

Parameter	Value
DIDO nodes	12
Maximum Control Torque (N*m)	± 0.025
Maximum Angular Rate (rad/s)	± 0.1
Maximum Quaternion q_1, q_2, q_3, q_4	± 1
Control Update Time (s)	real-time

Constraints for the maximum control torque were set to small values in order to reduce control discontinuities in DIDO’s control solution. Angular rates were set to

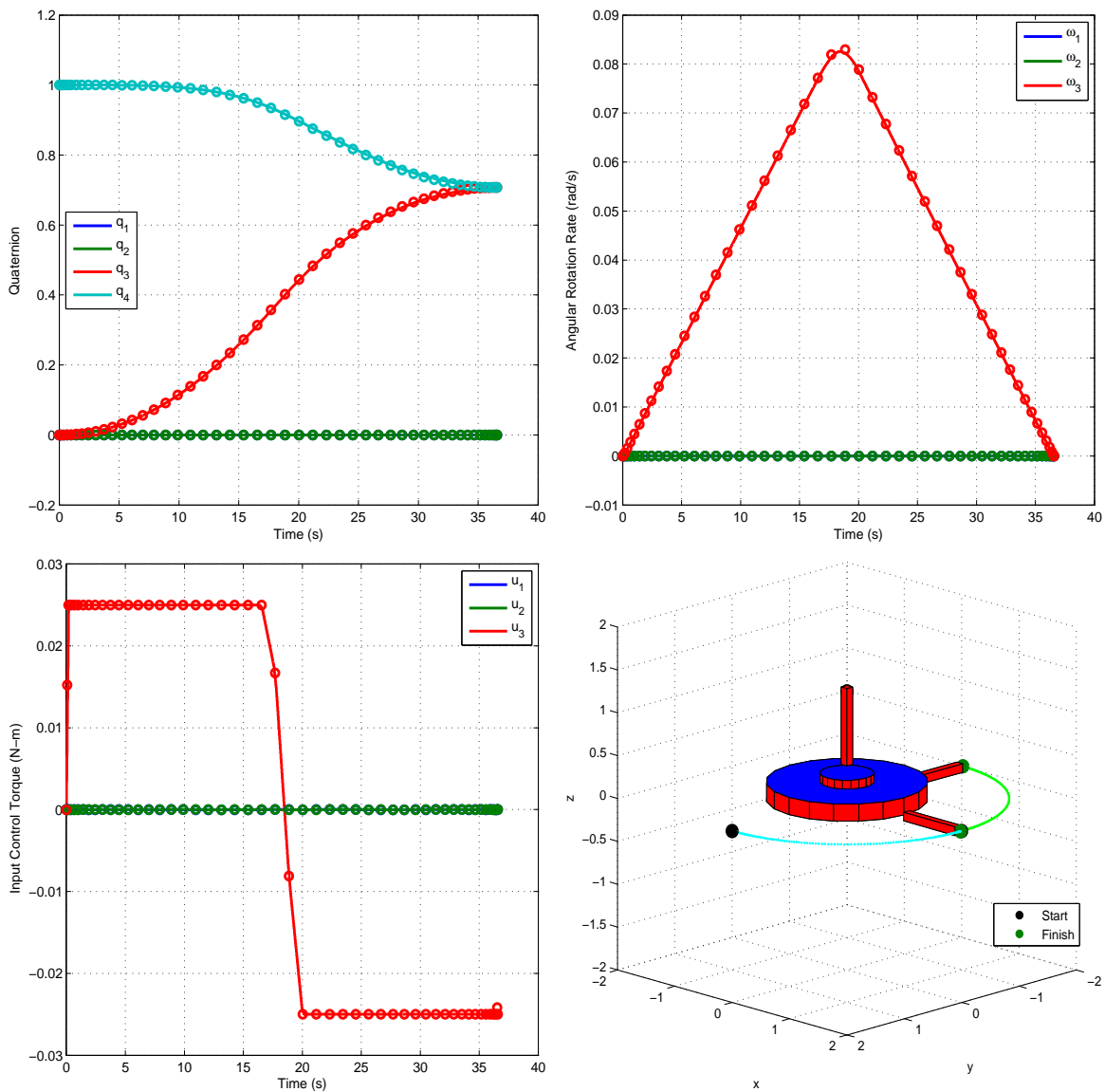


Figure 5.10: SimSAT II Simulation Results - EROC Controller 90° Maneuver

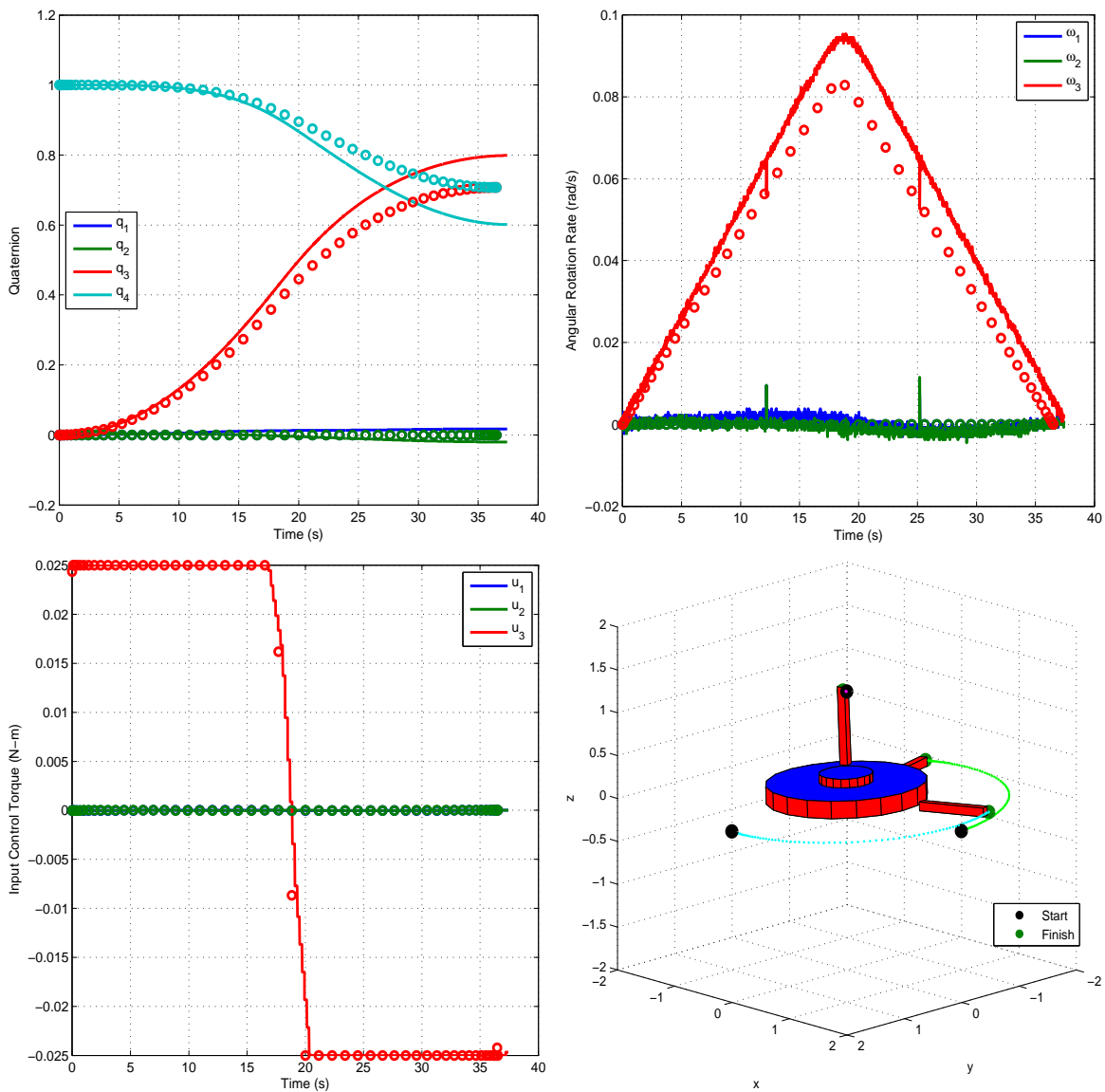


Figure 5.11: SimSAT II Experimental Results - EROC Controller 90° Maneuver

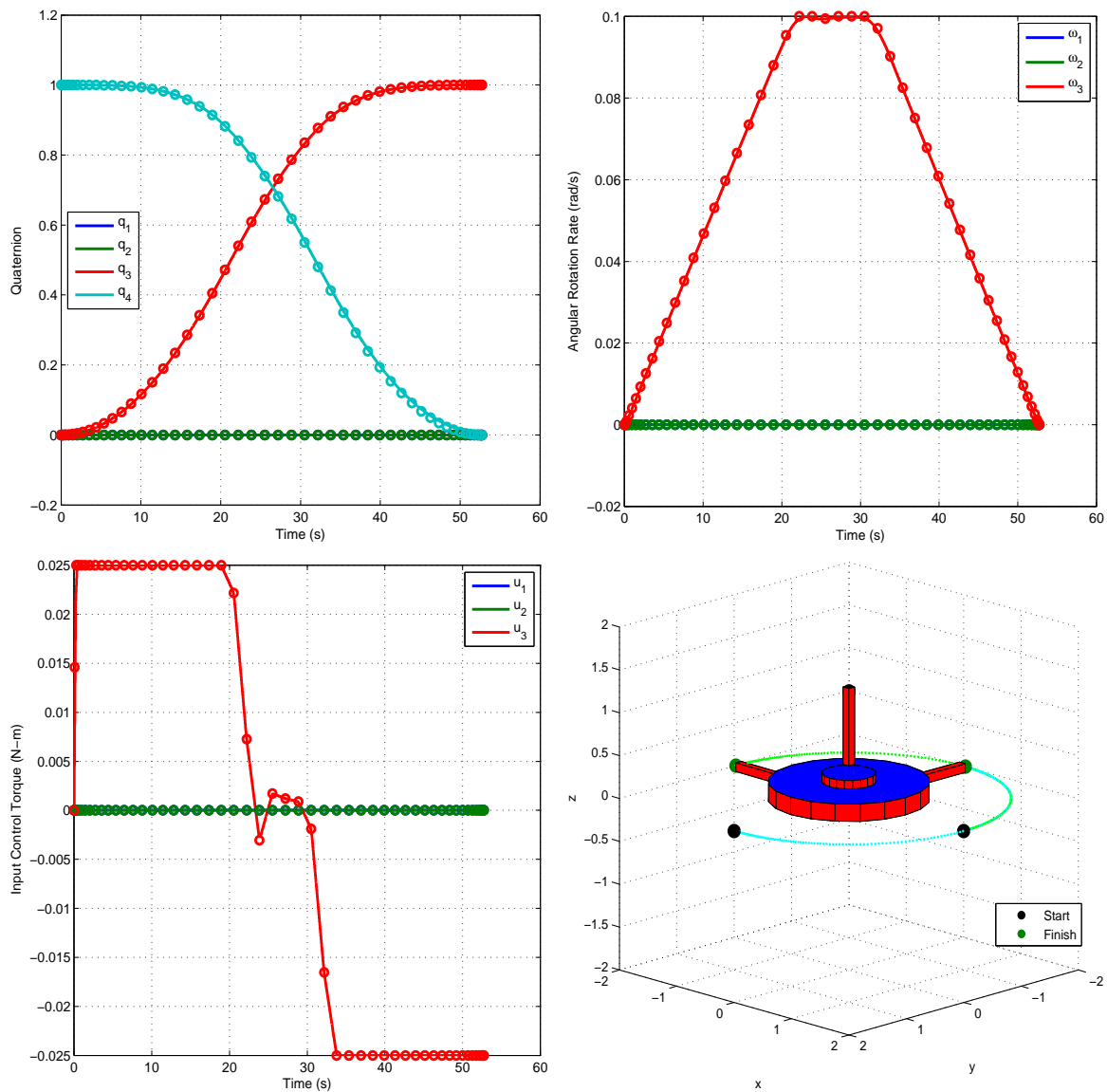


Figure 5.12: SimSAT II Simulation Results - EROC Controller 180° Maneuver

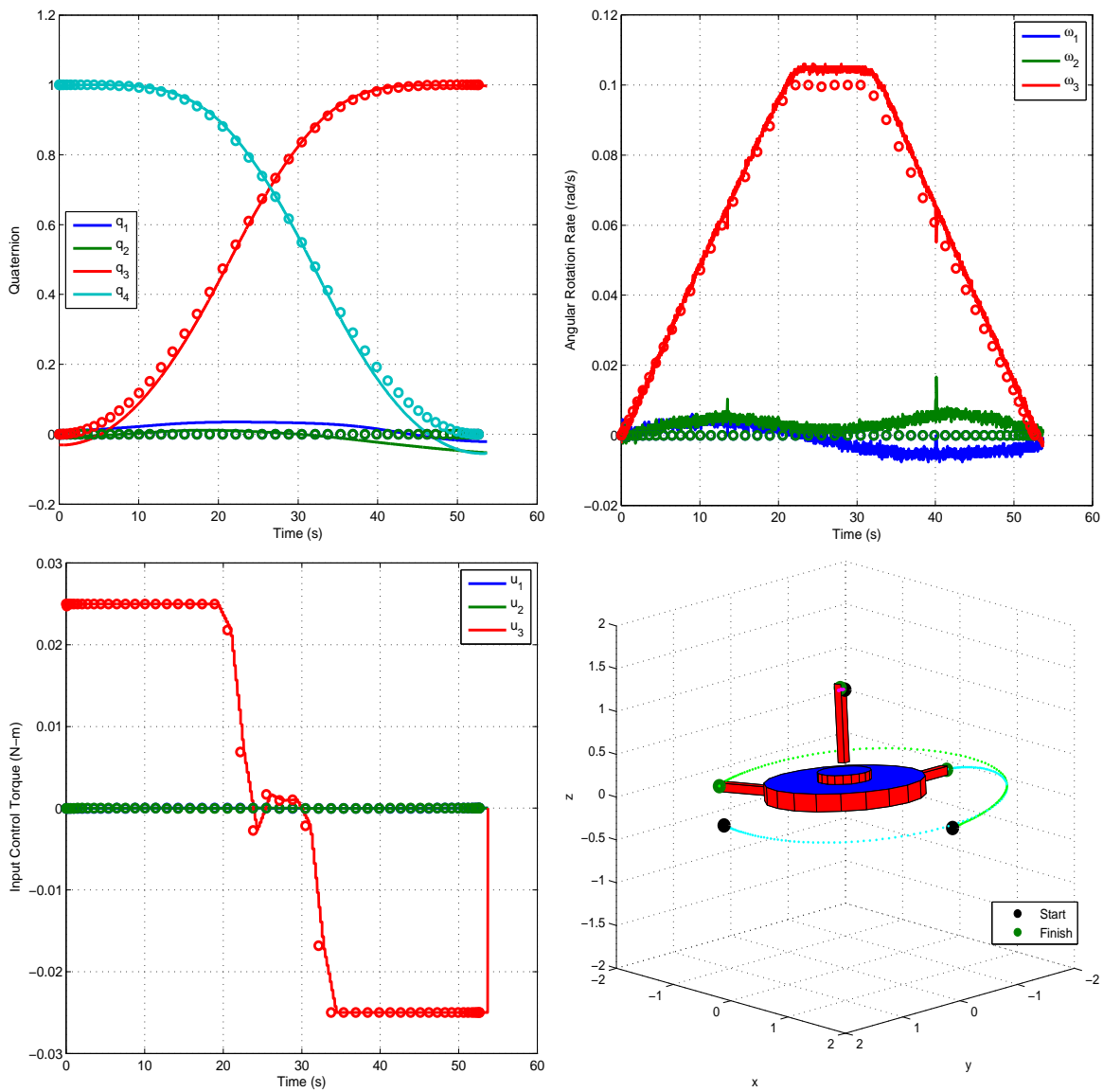


Figure 5.13: SimSAT II Experimental Results - EROC Controller 180° Maneuver

small values to reduce the effects of air drag. Using the 5th order polynomial coefficients listed in Table 4.1, a maximum angular rate of 0.1 rad/s will result in a maximum torque caused by air drag equivalent to $<6 \times 10^{-7}$ N-m. Even though the effects of drag are extremely small, they are still accounted for when calculating an optimal control solution. Unlike the traditional optimal control and eigenaxis maneuvers, all quaternion parameters for the RTOC maneuvers are unrestricted in order to better highlight the performance benefits of RTOC. A RTOC controller was not implemented on SimSAT II due to time constraints, therefore only model results for the RTOC controller will be presented.

5.3.1 Real-Time Optimal Control Model Results. Figure 5.2 shows the simulated results for the quaternion parameters and angular rotation rates of the SimSAT II using a RTOC optimal control maneuver for a 45° rotation yaw axis using real-time control updates. The RTOC controller results shown in Figure 5.14 clearly disagree with the DIDO estimated solution. The reason for this disagreement is shown in Figure 5.15 which demonstrates that by the second iteration of the RTOC controller computing a new optimal control solution that the propagated error in the model was greater than ϵ_{inner} error threshold of 0.1. Since the propagated error in the model was greater than the ϵ_{inner} error threshold, a new control solution is implemented at approximately 5 seconds into maneuver, causing a discontinuity in the control solution shown in Figure 5.14. Other discontinuities throughout the control solution are also due the propagated error being greater than the ϵ_{inner} error threshold.

The sharp discontinuity in the controls during the maneuver shown in Figure 5.14 are also due to DIDO finding an optimal control solution that will result in SimSAT completing the maneuver in even less time than the control solution originally calculated. As seen in Figure 5.16, starting at the third RTOC iteration the maneuver's final time gets faster every RTOC control iteration where propagated error was greater than ϵ_{inner} error threshold as shown in Figure 5.15.

Also shown in Figure 5.15 is that there were thirteen RTOC control iterations, but the total maneuver took approximately 25 seconds. Therefore, by the 13rd RTOC

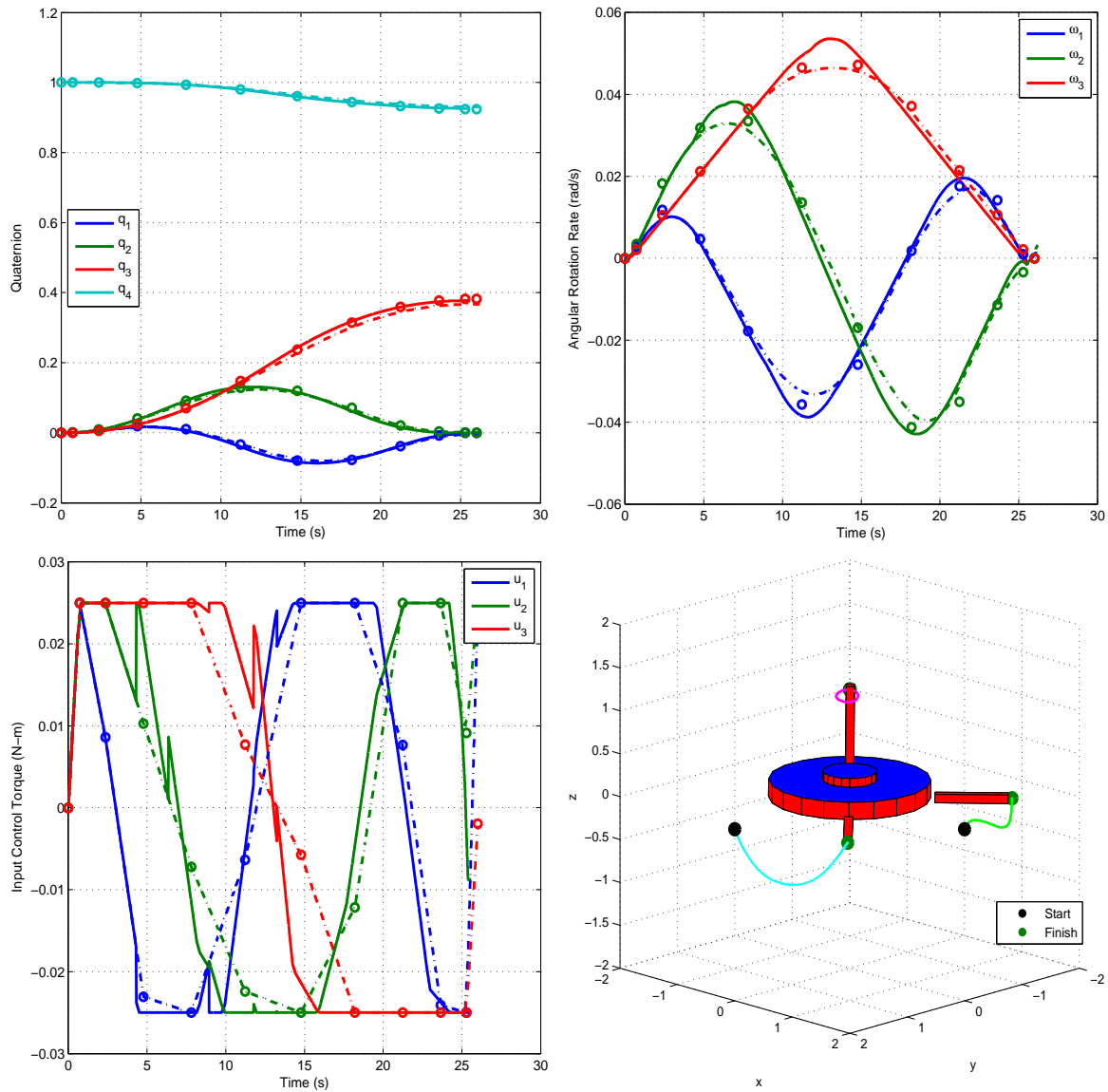


Figure 5.14: SimSAT II Model States - Real-time Update RTOC Controller 45° Maneuver

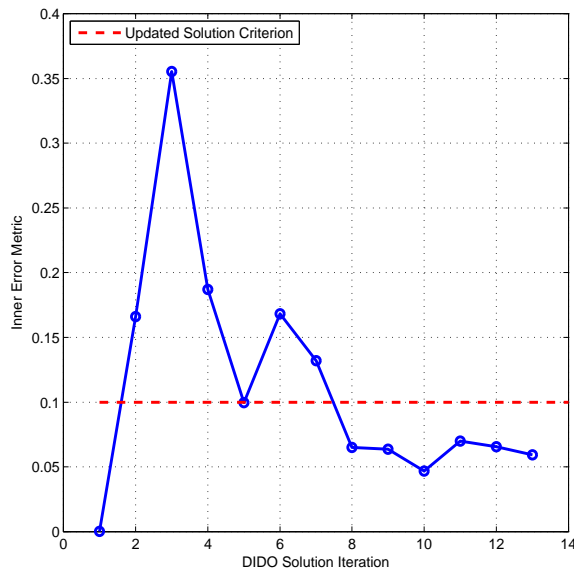


Figure 5.15: ϵ_{inner} - Real-time Update RTOC Controller 45° Maneuver

iteration, the error metric of “outer” control loop, calculated using Eq. (4.15), must have been smaller than the ϵ_{outer} error threshold of 0.005. Examining Figure 5.17 confirms this hypothesis for ϵ_{outer} . Figure 5.17 also shows that the “outer” control loop’s error metric decreases smoothly about each iteration. This occurs because the “outer” loop error metric is only a function of the quaternion error parameters that describe SimSAT II’s current orientation relative to the desired final orientation. This concept is described in more detail in Section 4.3. Since the quaternion parameters for the entire RTOC maneuver smoothly change from their values to the values describing the desired final orientation, as shown in Figure 5.14, the fact that the “outer” loop error metric decreases smoothly should come as no surprise.

Conversely, the “inner” error metric for this maneuver shown in Figure 5.15 displays what appears to be erratic variation, however, as it turns out, the variation is rather predictable. This partly occurs because it is a function of both quaternions and angular velocity, as described in Section 4.3. However, it also occurs because the RTOC controller is implementing a new control solution each time the “inner” error metric exceeds the ϵ_{inner} error threshold value. Figure 5.15 demonstrates that with the exception of the third and eighth RTOC iteration, every time the “inner” error metric exceeds the ϵ_{inner} error threshold value that the “inner” error metric at the next RTOC iteration was always

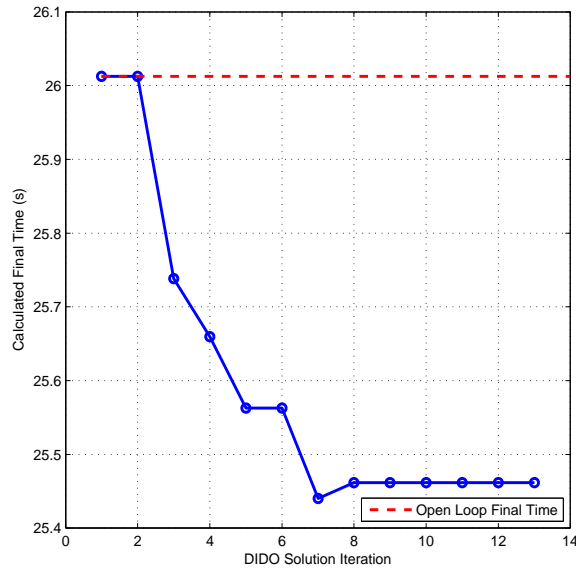


Figure 5.16: SimSAT II Simulation - Real-time Update RTOC Controller 45° Maneuver Final Time

smaller. This occurred because the RTOC controller was “correcting” the maneuver similar to a regulator steering state error to zero. However, the RTOC controller does not “correct” the maneuver by steering states towards its previous optimal path because that path may no longer be optimal. Instead, the RTOC controller calculates a new optimal path using current measured states and continues to complete the maneuver.

RTOC optimal control model performance results for 90° and 180° yaw axis rotation maneuvers are shown in Figures 5.18 and 5.19. The results of these maneuvers are characterized by the errors sources highlighted for the 45° yaw axis rotation maneuver shown in Figure 5.14. Table 5.5 compares the results of the RTOC controller performance indices with the performance indices for both the traditional optimal control and eigenaxis maneuvers described in Section 5.2. As with comparison tables, the units of the performance indices are in seconds since all of the performance indices minimize a maneuver’s final time.

Table 5.5: Optimal Control Performance Index Comparison

Rotation	RTOC PI (s)	OLOC PI (s)	Δt	EROC PI (s)	Δt
45°	25.40	25.00	+1.57%	25.83	-1.69%
90°	34.89	35.41	-1.49%	36.53	-4.70%
180°	44.26	50.92	-15.05%	52.65	-18.96%

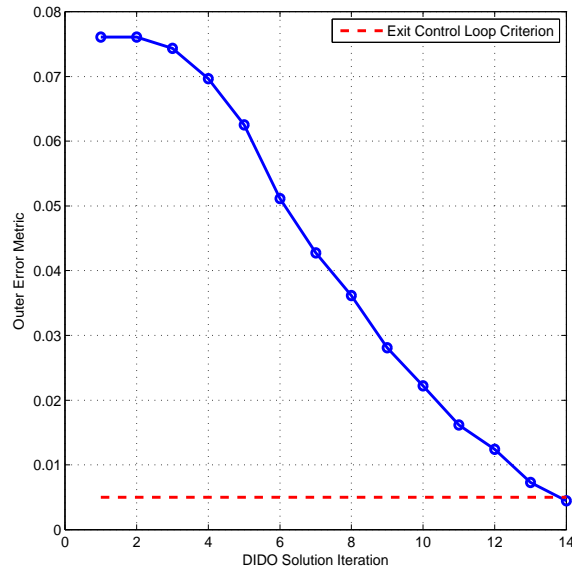


Figure 5.17: ϵ_{outer} - Real-time Update RTOC Controller 45° Maneuver

Clearly, the RTOC control provides significant improvement in slew time during longer maneuvers. The comparison in Table 5.5 is not necessarily an equal comparison because the RTOC controller uses 12-node DIDO solutions, in order to calculate a new optimal solution as fast as possible, while the OLOC and EROC controllers both use 50-node DIDO solutions. Additionally, the RTOC maneuvers are not subject to the q_1 and q_2 constraints that prevent rotation about roll and pitch axes. Not using these rotational constraints is not a bad assumption because a satellite in space would not have rotational limitations due to an air bearing. Figure 5.20 shows the negative effects of the q_1 and q_2 constraints on the 50-node DIDO solutions for the 90° and 180° maneuvers using an OLOC controller. In both graphs, the q_1 and q_2 solution travel linearly along the ± 0.1 boundary constraint, thus suggesting that the unrestricted optimal solution lies outside of the ± 0.1 bounds. This means that the OLOC controller could have completed the maneuver by as much as 13% faster for the 180° maneuver had there been no constraints for q_1 and q_2 . For the 45° maneuver, the RTOC controller was slight slower than the OLOC controller. This occurred because the RTOC controller only used 12 LGL nodes to calculate an optimal control solution, while the OLOC controller used 50 LGL nodes and therefore the RTOC controller had more error in its optimal control solution. The

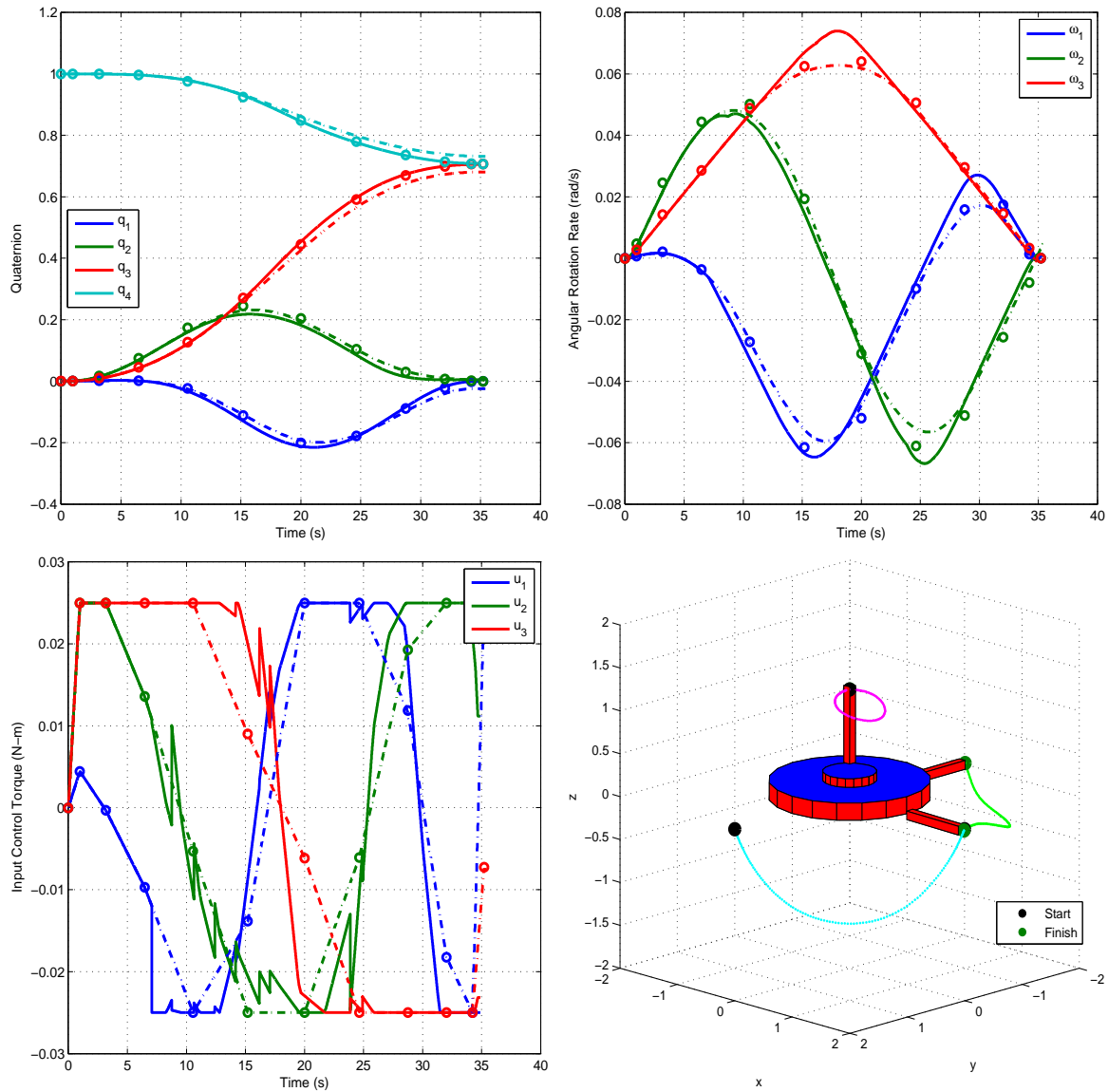


Figure 5.18: SimSAT II Simulation - Real-time Update RTOC Controller 90° Maneuver

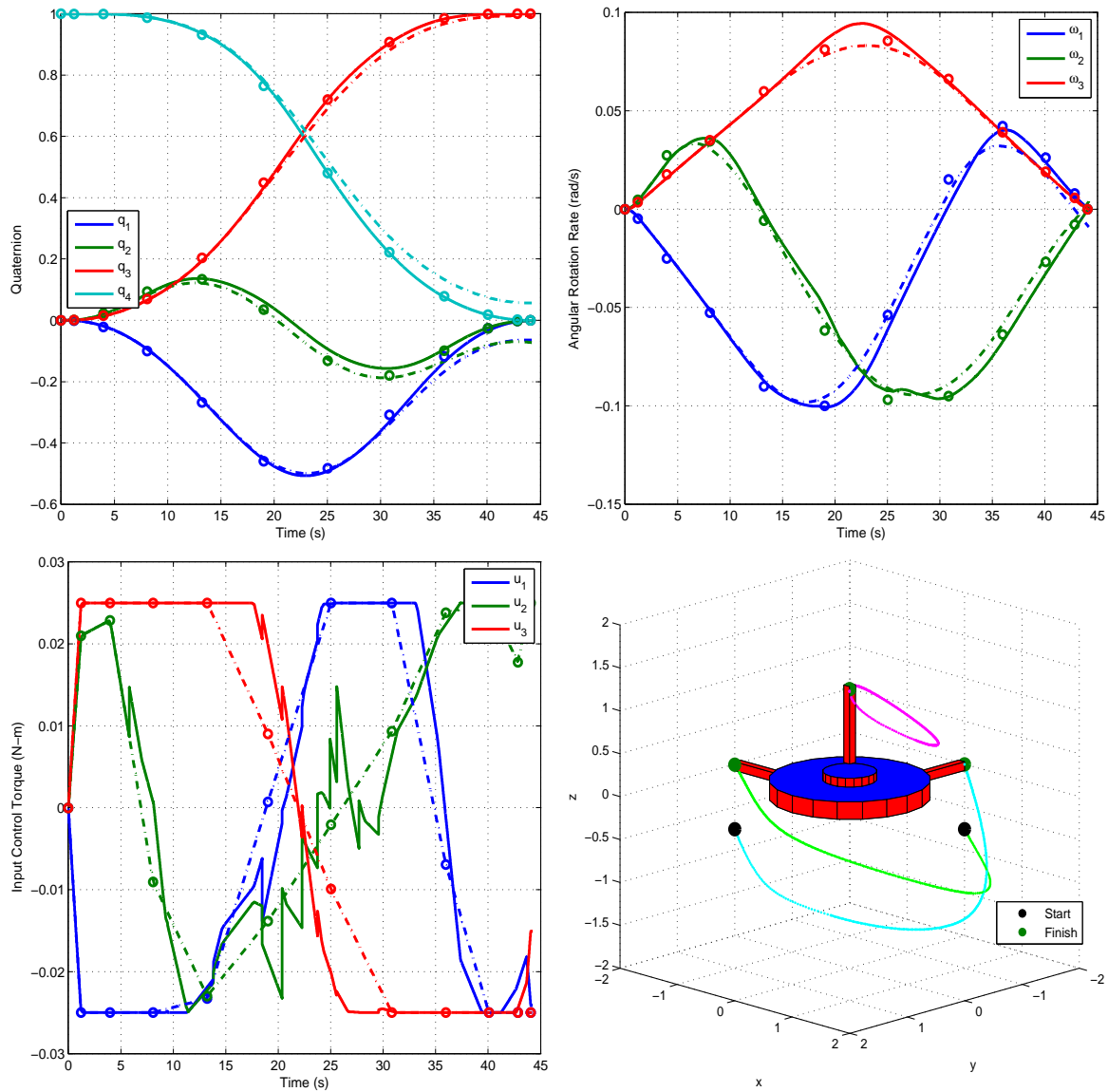


Figure 5.19: SimSAT II Simulation - Real-time Update RTOC Controller 180° Maneuver

effects of the number of LGL nodes has on an optimal control solution calculated by DIDO is discussed in the next section.

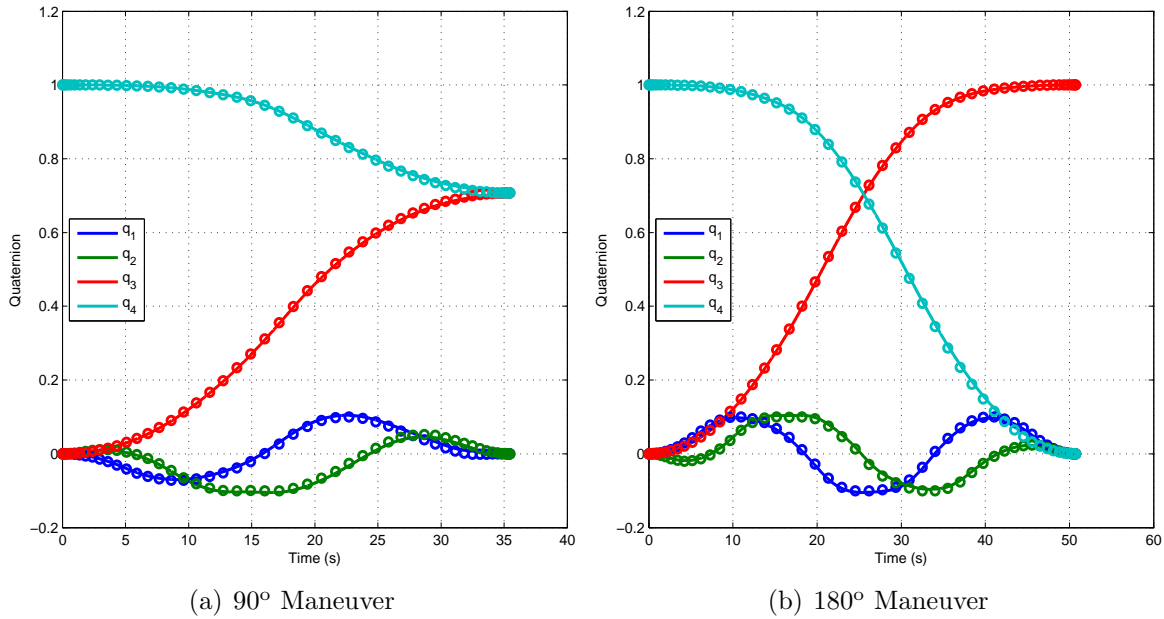


Figure 5.20: Quaternion States - OLOC Controller Maneuvers Using Quaternion Boundary Constraints

5.4 Optimal Control Parameter Variation

This section explores how parameter variation in problem formulation can affect the results of SimSAT II’s OLOC and RTOC controllers. First, this section will show how the results of pseudospectral-based optimal control solving tools such as DIDO are sensitive to the number of LGL nodes requested during problem formulation. The next section will demonstrate how a RTOC controller will perform as the time between the implementation of new optimal control solutions is increased. The following section will demonstrate how the results found by DIDO have been shown to vary from one function call to the next, without changing the formulation of the problem. The subsequent section will demonstrate how using an accurate “guess” for what the optimal control solution should be will reduce the variation in optimal control solutions found by DIDO. The last section demonstrates how parametric uncertainty in SimSAT II’s MOI matrix can affect a traditional optimal controller, and how these effects can be overcome using a RTOC controller.

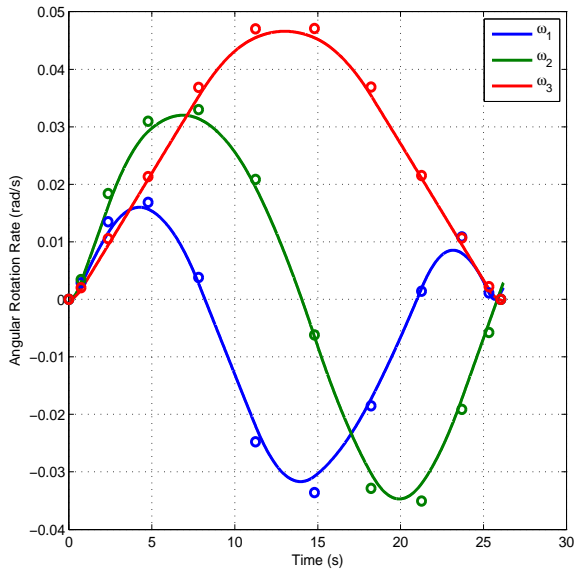
5.4.1 *Optimal Control Solution Accuracy.* The accuracy of optimal control solutions found using DIDO is highly dependent on the number of LGL nodes requested during formulation of the problem. To demonstrate this effect, SimSAT II’s OLOC controller is simulated the same 45° yaw axis rotation maneuver using a 12-node, 25-node, and a 50-node DIDO solution. The angular rotation rate histories of the propagated 12-node, 25-node, and a 50-node OLOC controller solutions are compared in Figure 5.21. The graph on the left in Figure 5.21 is the 12-node DIDO solution, the graph on the right is the 25-node DIDO solution and the graph on the bottom in is the 50-node DIDO solution. Clearly, the propagated results of the 50-node optimal control solution match closer to the DIDO calculated solution than the 12-node and the 25-node optimal control solutions. This occurred, because as the number LGL nodes used by DIDO increases, the more “bang-bang” like the resulting controls appear. Figure 5.22 shows the control solutions corresponding to the angular rotation rates shown in Figure 5.21, where the graph on the left is the 12-node DIDO control solution, the graph on the right is the 25-node DIDO control solution and the graph on the bottom in is the 50-node DIDO control solution. The 50-node control solution in Figure 5.22 more closely resembles a “bang-bang” control solution than either the 12 or 25-node solutions.

The optimality of the 12-node, 25-node, and a 50-node DIDO solutions was analyzed by comparing their performance indices. Table 5.6 demonstrated that the optimality of DIDO’s control solution was increased by up to 4.6% through increasing the number of nodes used by DIDO for the calculation.

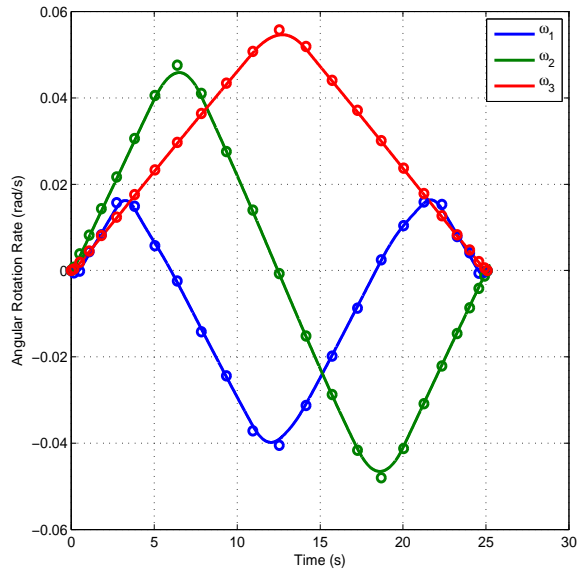
Table 5.6: DIDO Node Versus Performance Index Comparison

DIDO Nodes	Performance Index (s)	Percent Change
12	26.2	-
25	25.2	-3.8%
50	25.0	-4.6%

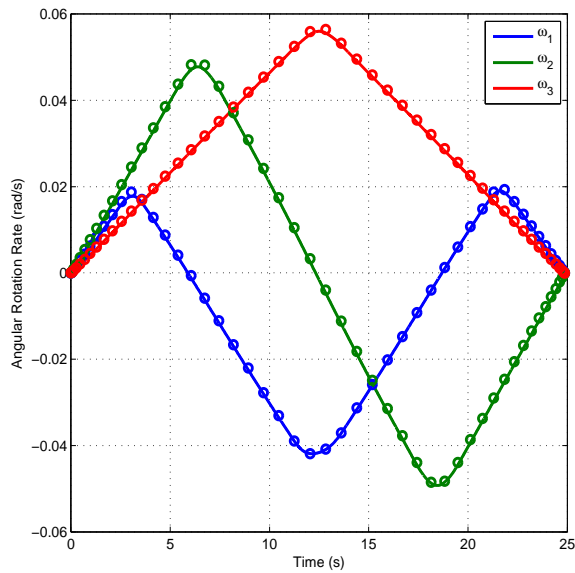
The drawback of increasing the nodes is that DIDO takes longer to compute an optimal control solution. Table 5.7 shows that the DIDO’s required computation time significantly increases as the number of nodes increases, and thus the accuracy of the solution. Using too many nodes to calculate an optimal control solution, can cause a



(a) 12 Nodes



(b) 25 Nodes



(c) 25 Nodes

Figure 5.21: Angular Rate Simulation - LGL Node Variation for OLOC Controller 45° Maneuver

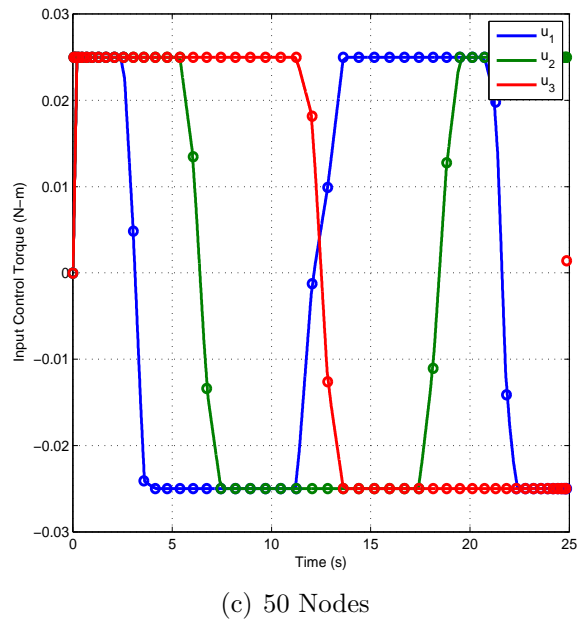
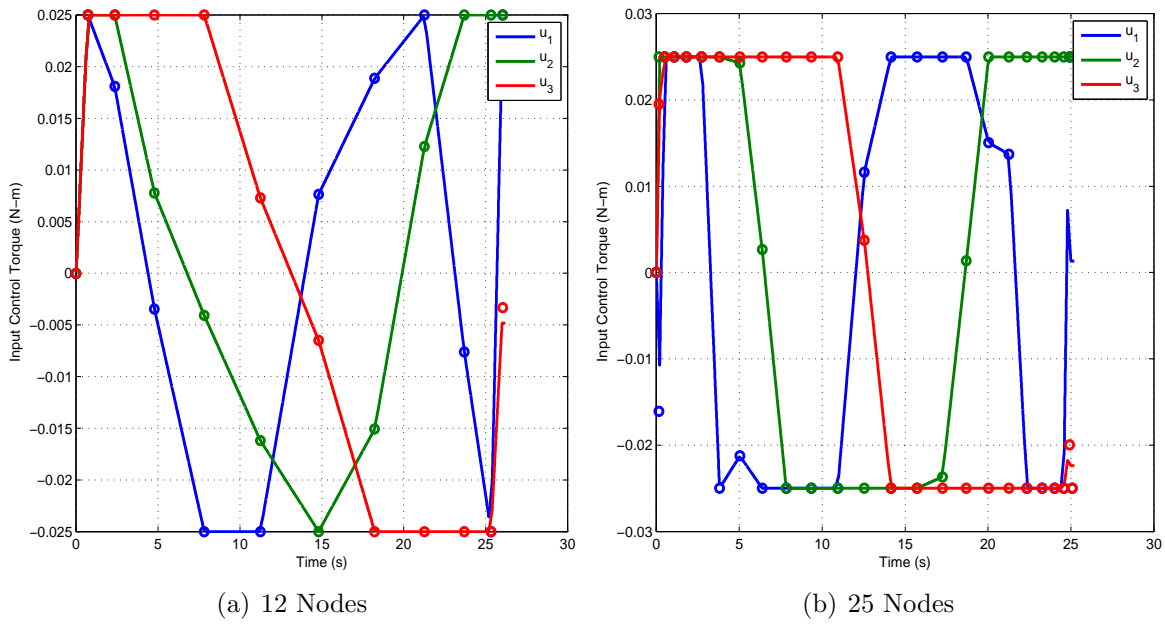


Figure 5.22: Control Simulation - LGL Node Variation for OLOC Controller 45° Maneuver

RTOC controller to behave like an OLOC controller. For instance, if it takes SimSAT II 25 seconds to complete a 45° yaw axis slew, and it takes 26.11 seconds to calculate a new 50-node control solution, then a RTOC controller would only be able to complete one control iteration before completing the maneuver.

Table 5.7: DIDO Node Versus Solution Calculation Time Comparison

DIDO Nodes	Solution Calculation Time (s)	Percent Change
12	3.28	-
25	10.67	+225.3%
50	26.11	+696.0%

5.4.2 RTOC Solution Update. As discussed in the previous section, if the number of nodes used in DIDO’s optimal control solution is too high, it will decrease the number control iterations that a RTOC controller can perform during the course of a maneuver. This section will analyze the performance of a RTOC controller for a 45° yaw axis slew maneuver as the time between updating new control solutions is increased. For each model, a RTOC controller is used to execute a 12-node DIDO optimal control solution that will be updated every 1 second, 5 seconds, and in “real-time.” The term “real-time” refers to updating the optimal control solution immediately following DIDO computing the previous optimal control solution. As shown in the previous section, 12-node DIDO optimal control solutions are more prone to error than higher nodal solutions, therefore, the RTOC controller will need to update the optimal control solution several times during the course of the maneuver.

The results of varying the optimal control update time for a RTOC control is shown in Figure 5.23, where the graph on the left uses 1 second optimal control updates, the graph on the right uses 5 second optimal control updates, and the graph on the bottom using “real-time” optimal control updates.

The 5-second update RTOC controller results shown on the right of Figure 5.23 misses the desired final angular rotation rates. In fact, the 5 second update RTOC controller performs worse than the traditional optimal controller. This occurs because of the updated control solutions are 5 seconds out of phase with SimSAT II’s current

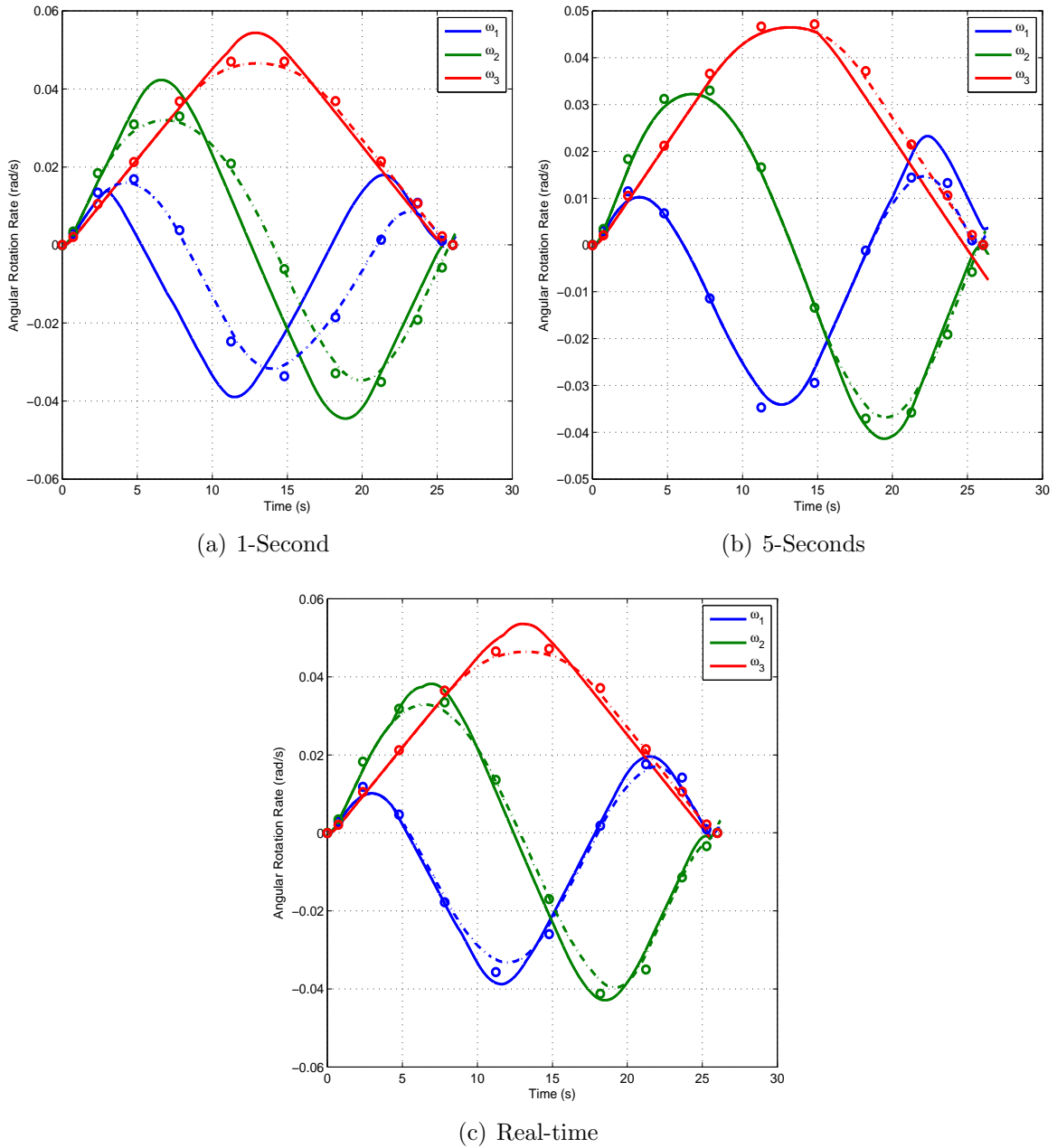


Figure 5.23: Angular Rate - Update Time Variation for RTOC Controller 45° Maneuver

states. The out of phase controls are always driving SimSAT II to an optimal path that was computed 5 seconds previously. If the new optimal controls cause SimSAT II to deviate from the optimal path, it will take up to 5 seconds to find a new optimal path which will also be out of phase. Correcting deviations from the optimal path as SimSAT II approaches the end of a maneuver because difficult, if not impossible, as evident by the 5-second update RTOC controller results.

The 1-second and “real-time” update RTOC controller results shown in Figure 5.23 both converge to the desired final angular rotation rates. This suggests that a phase lag for these controllers is not too large that it cannot be corrected, as was the case with the 5-second update RTOC controller. The 1-second update RTOC controller updated its control solutions every one second, but the “real-time” update RTOC controller only updated its control solutions based on DIDO’s computation time. Figure 5.24 shows how DIDO’s computation time varied with each optimal control iteration. While the DIDO calculation time is not very consistent, it does show that as SimSAT II approaches the desired final states it takes DIDO less time to compute an optimal control solution. This suggests that the number of nodes used by DIDO near the end of the maneuver may be increased to maintain a consistent computation time during the course of the maneuver and increase the accuracy of the optimal control solutions.

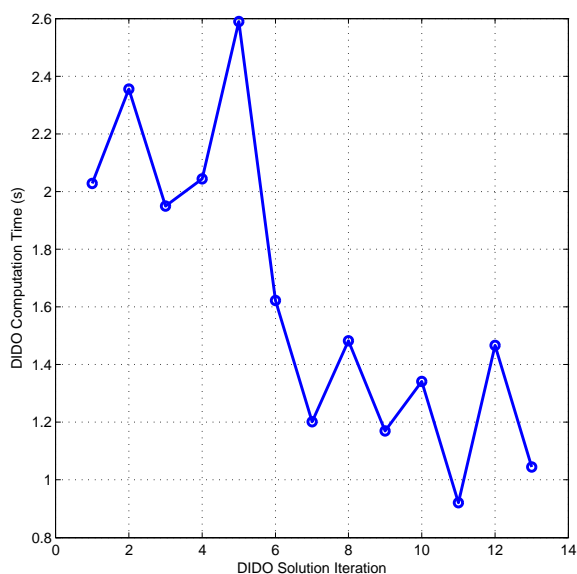


Figure 5.24: DIDO Computation Time - 45° Real-time Update RTOC Maneuver

5.4.3 *Random Optimal Control Solution Variation.*

Infrequently, DIDO will compute different solutions for the same problem despite there being no difference in the formulation of the problem. Figure 5.25 show the angular rotation rates for the same 45° yaw axis rotation using a RTOC real-time update controller and the lower two graphs track the change in ϵ_{inner} for the angular rotation rate optimal solution above it.

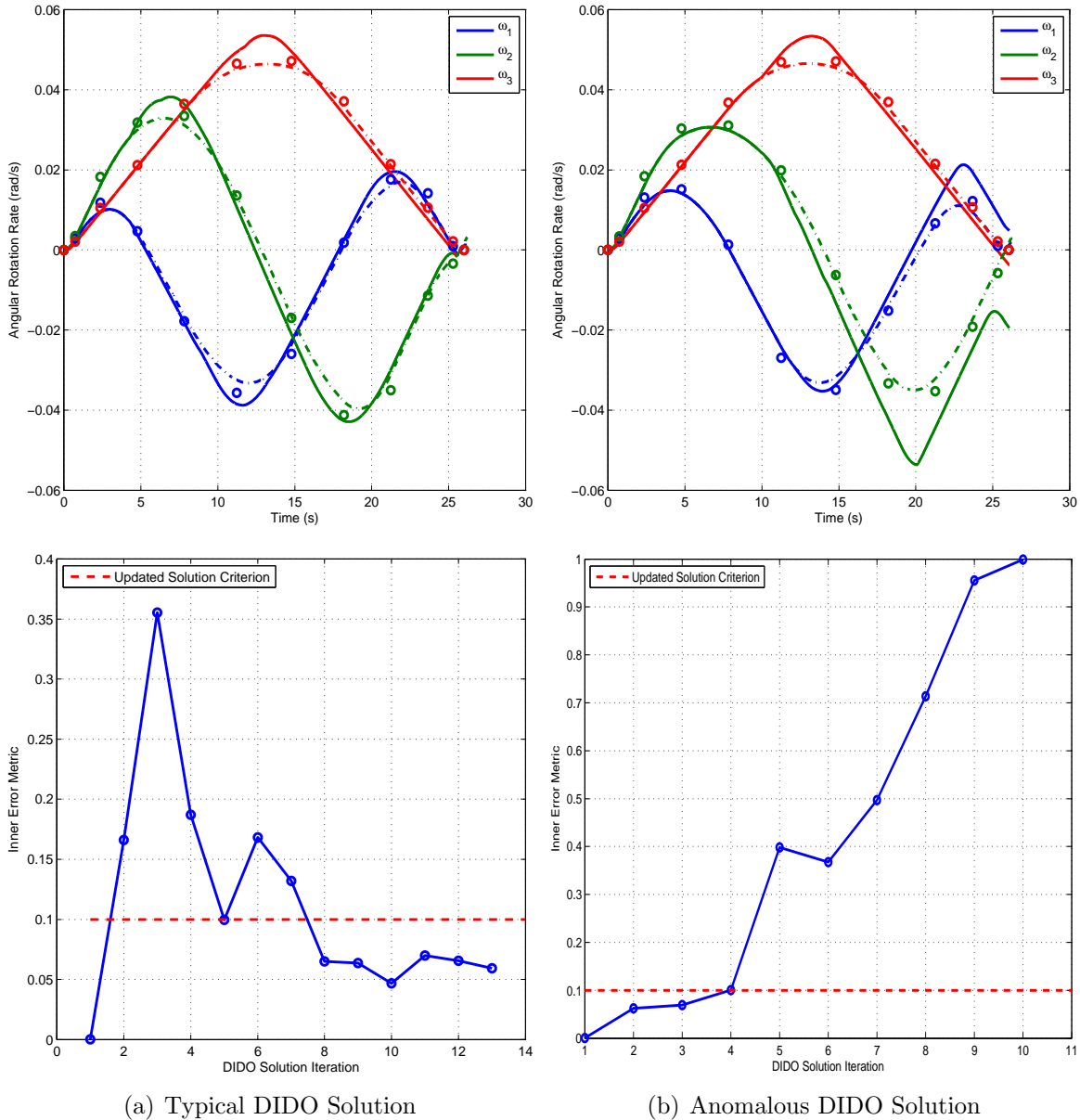


Figure 5.25: Different Optimal Solutions - Real-time Update RTOC Controller 45° Maneuver

Due to the time constraints of this thesis, detailed analysis was not completed for this anomaly. Preliminary analysis analyzed the change in ϵ_{inner} during each DIDO

iteration. As discussed in Section 5.3.1 it is expected that ϵ_{inner} will decrease following where ϵ_{inner} was greater than the error threshold of 0.1. However, for the angular rotation rate graph on the right, which did not converge on the correct solution, ϵ_{inner} almost always increased with each iteration of the RTOC controller, suggesting that the solutions found by DIDO and being implemented by SimSAT II are non-optimal solutions.

The first step for future investigation of this anomaly is to track and analyze the optimality of control solutions found by DIDO by graphing the problem's Hamiltonian H from Eq. (2.64) over the course of the maneuver. For optimal control problems that can be solved analytically, H will remain constant over the course of an optimal control maneuver. For optimal control problems solved by pseudospectral methods such as those used by DIDO, there will be oscillations in H because DIDO estimates the problem using a Lagrange interpolating function. The magnitude of these oscillations will be dependent on the number of nodes DIDO uses to estimate the problem, where DIDO will have a higher fidelity estimation the problem with more nodes. Hard and steadfast guidelines do not exist for an acceptable amount of oscillation of the H , however variations in H much larger 0.1 suggest that the optimal solution found by DIDO may not be the optimal solution to the problem. Figure 5.26 shows an example of a Hamiltonian output as part of a DIDO calculated solution to an optimal control problem.

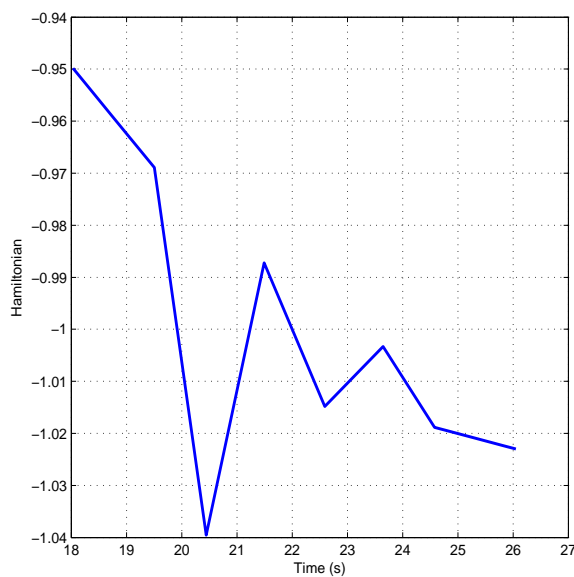


Figure 5.26: DIDO Optimal Control Solution Output - Hamiltonian H

Further analysis of this anomaly would include tracking to DIDO's control output to observe whether or not DIDO calculates or implements an infeasible control solution during the maneuver and doing more detailed analysis of the problem itself to ensure that it is properly formulated. No matter what the reason, variation in optimal control solution results like that depicted in Figure 5.25 could result in total loss of a spacecraft. Therefore, record anomalous results must be further analyzed before implementing a RTOC controller on an actual system.

5.4.4 Use of Optimal Control Solution Guess. Optimal control solutions calculated by DIDO are sensitive to parameter variation in problem formulation. Even minor changes in the initial states of the maneuver can result in significant changes in the optimal solution. This can be especially problematic for an RTOC controller which is constantly calculating and recalculating the optimal solution to the problem. One way to increase the consistency of DIDO's calculated optimal control solutions is to use a "guess" during problem formulation [41]. A "guess" consists of a discrete set of states that are believed to be located along the optimal control solution. In the case of a RTOC controller, the "guess" used for each optimal control solution iteration is the previously calculated optimal control solution that has been truncated to account for DIDO's previous calculation time.

Figure 5.27 shows the propagated angular rotation rate solution to the same 1-second update RTOC controller simulation of a 45° yaw axis rotation, except the graph on the left uses a "guess" during optimal control solution iteration and the graph on the right does not use a "guess." When not using a "guess," the optimal control solutions calculated by DIDO tend to change direction very sharply and as a result of the RTOC controller implementing optimal control solutions that are less optimal than the preceding optimal control iteration. Corresponding to the simulation results shown in Figure 5.27, Figure 5.28 compares the changes in the optimal control solution's final time after each iteration, where the graph on the left corresponds to the RTOC controller using a "guess" and the graph on the right corresponds to the RTOC controller not using a guess.

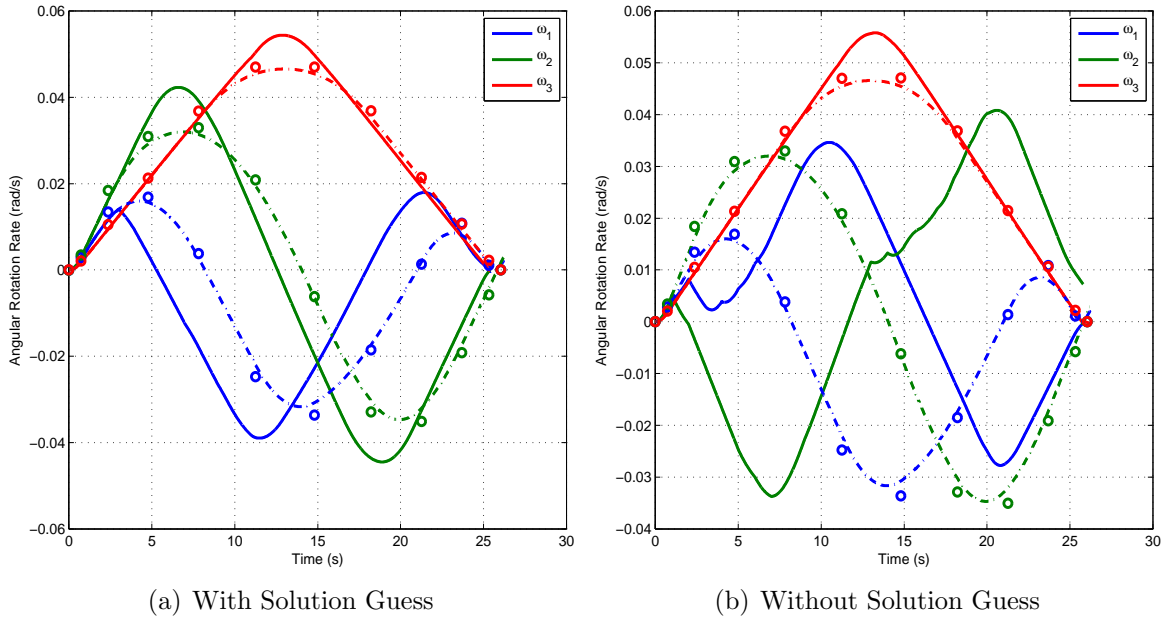


Figure 5.27: Angular Rotation Rate Simulation - Solution Guess Variation for 1s Update RTOC Controller 45° Maneuver

The RTOC controller not using a “guess” sometimes implements an optimal control solution that is less optimal than the previous iteration, while the RTOC controller using a “guess” always implements an optimal control solution that equal to or more optimal than the previous iteration. At the nodes where the RTOC controller implements a less optimal control solution in Figure 5.28 corresponds to a sharp change in direction in the angular rotation rate as seen in Figure 5.27.

5.4.5 Effects of MOI Parametric Uncertainty. Sudden changes in the MOI matrix can cause traditional closed-loop feedback controllers to become unstable and possible result in the loss of a spacecraft. If the MOI matrix used by an OLOC controller to calculate an optimal control solution for SimSAT II is different than SimSAT II’s actual MOI matrix, the OLOC optimal control solution may no longer be optimal and may even lead SimSAT II off course.

Figure 5.29 shows simulation testing of the effects of SimSAT II’s actual MOI matrix being 10% greater than the MOI matrix used by SimSAT II’s OLOC controller to calculate an optimal solution to a 180° yaw axis rotation. At the end of the maneuver,

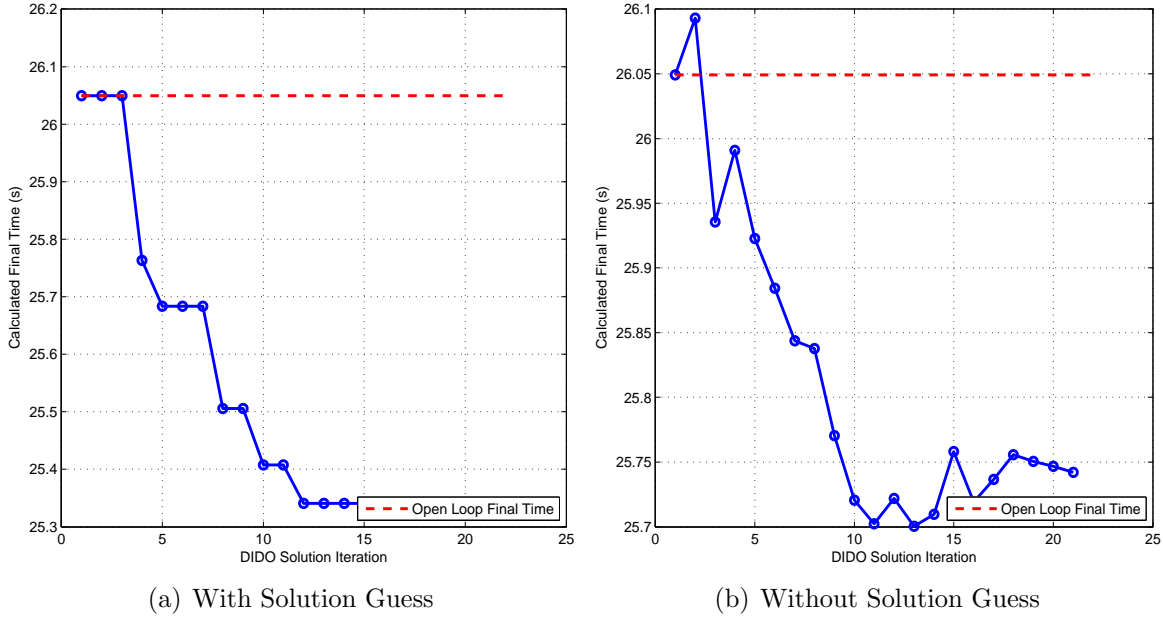


Figure 5.28: Maneuver Final Time - Solution Guess Variation for 1s Update RTOC Controller 45° Maneuver

all of the quaternion parameters and angular rotation rates have significant error. The error in the final states can be significantly reduced at the cost of time optimality.

Figure 5.30 shows the simulation testing of the effects of SimSAT II’s actual MOI matrix being 10% greater than the MOI matrix used by SimSAT II’s real-time update RTOC controller to calculate an optimal solution to a 180° yaw axis rotation. The RTOC controller is continuously calculating new optimal control solutions using the erred MOI matrix causing error between the estimated DIDO solution and the propagated solution to be greater than the ϵ_{inner} error threshold as shown in Figure 5.31. Therefore, a new optimal control solution is implemented at every control iteration and “corrects” the state errors caused by the MOI matrix. The new optimal control solution is also incorrect because it was calculated using the wrong MOI matrix as well, however as SimSAT II gets closer to the end of the maneuver the effects and reduces its angular rotation rate, the effects of the MOI errors are also reduced due to SimSAT II’s EOM shown in Eq. (4.5).

As the effects of the MOI are reduced, the RTOC controller can make adjustments to the controls to complete the maneuver by reducing the ϵ_{outer} error less than the

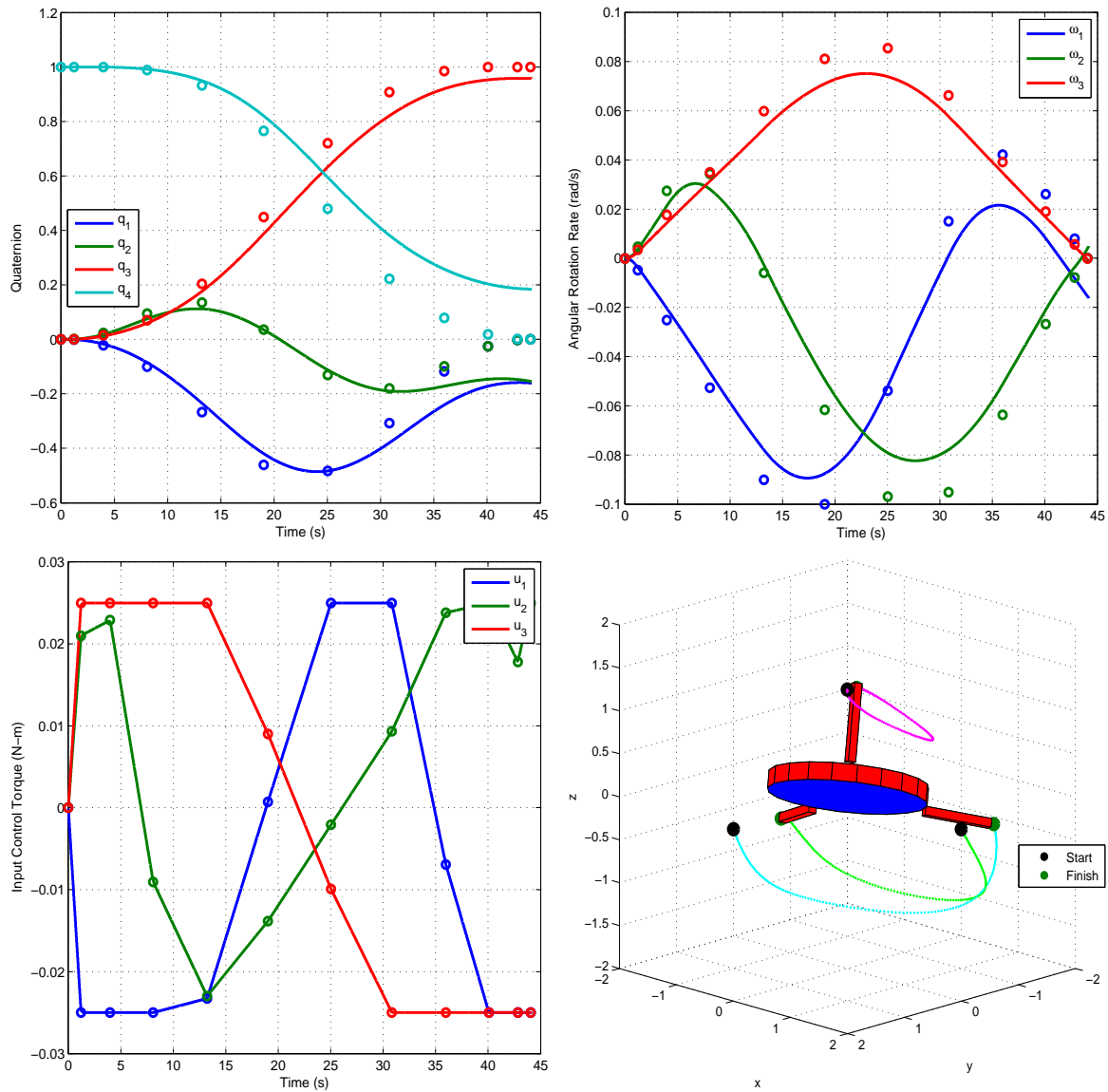


Figure 5.29: SimSAT II Simulation - 10% MOI Variation for Real-time Update RTOC Controller 180° Maneuver

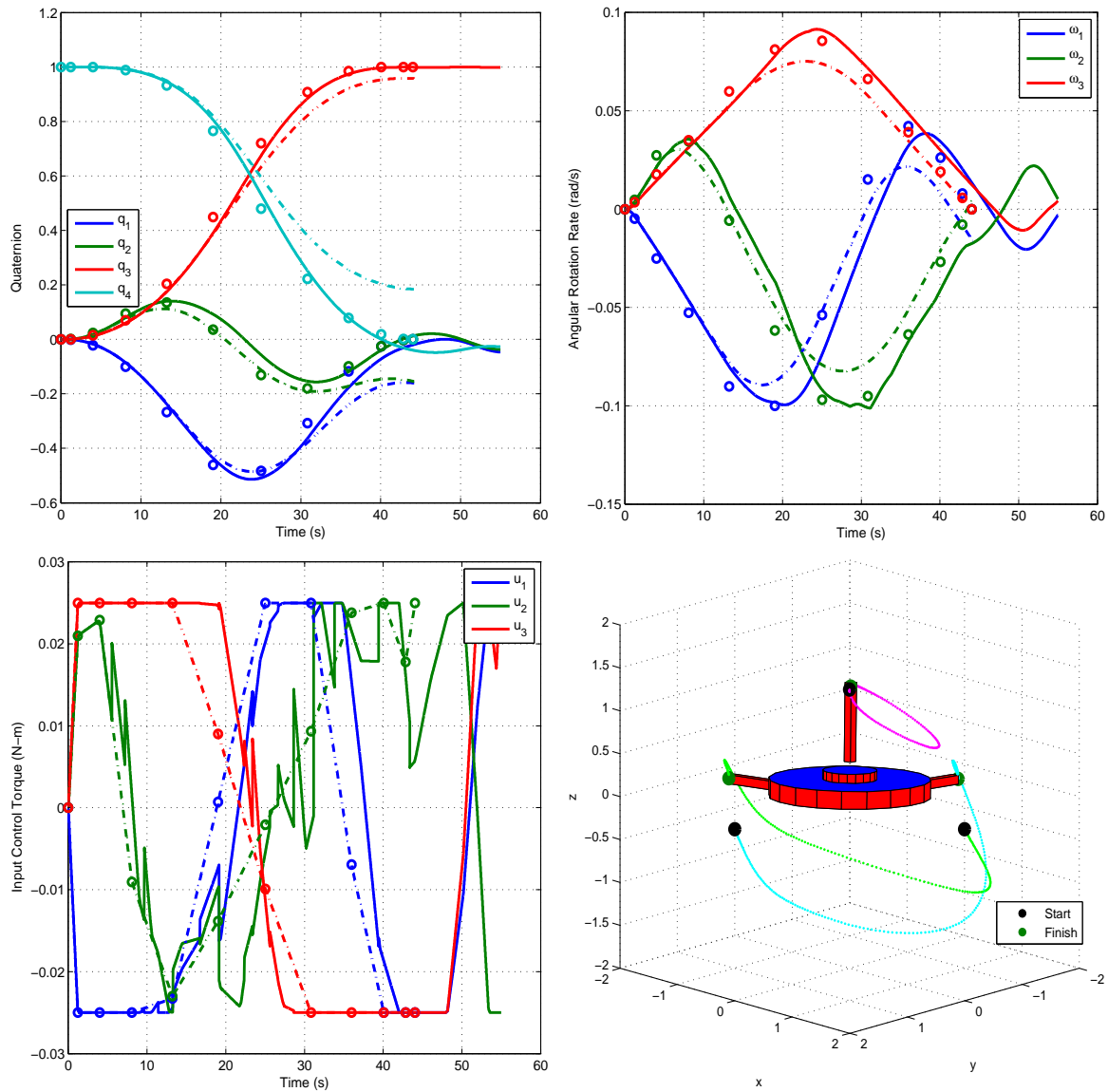


Figure 5.30: SimSAT II Simulation - 10% MOI Variation for Real-time Update RTOC Controller 180° Maneuver

threshold value. As shown in Figure 5.30, the RTOC controller gets close to completing the maneuver by the original maneuver final time, however it overshoots the final states and corrects itself until it meets the ϵ_{outer} error threshold requirement. This behavior resembles overshooting and oscillatory steady-state behavior of a closed-loop feedback controller, which suggests that it may be possible to manipulate the ϵ_{outer} and ϵ_{inner} error metric parameters to change the RTOC controllers performance, similar to how the gain matrix of a closed-loop feedback controller can be changed.

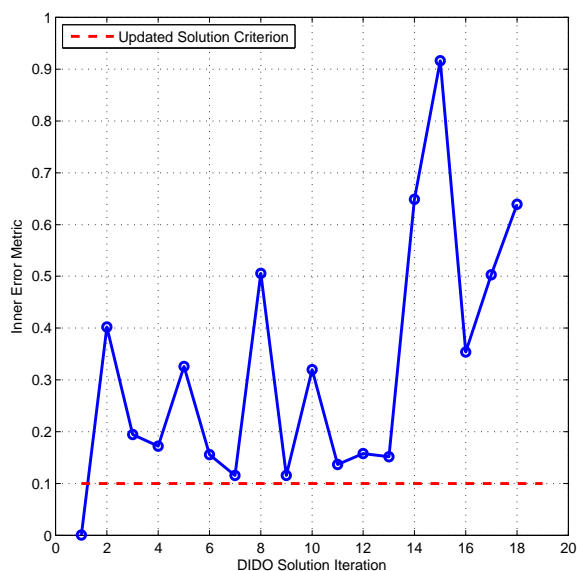


Figure 5.31: ϵ_{inner} - 10% MOI Variation for Real-time Update RTOC Controller 180° Maneuver

Figure 5.32 shows how the final time for the maneuver changes with each optimal control solution iteration of the RTOC controller. Up until the thirteenth iteration, the RTOC controller's final time for the maneuver changed $<3\%$ from the originally calculated final maneuver time. However, there is a $>40\%$ change in final time for the maneuver from iteration thirteen to eighteen. This sharp increase corresponds to the RTOC controller overshooting the final state and correcting itself.

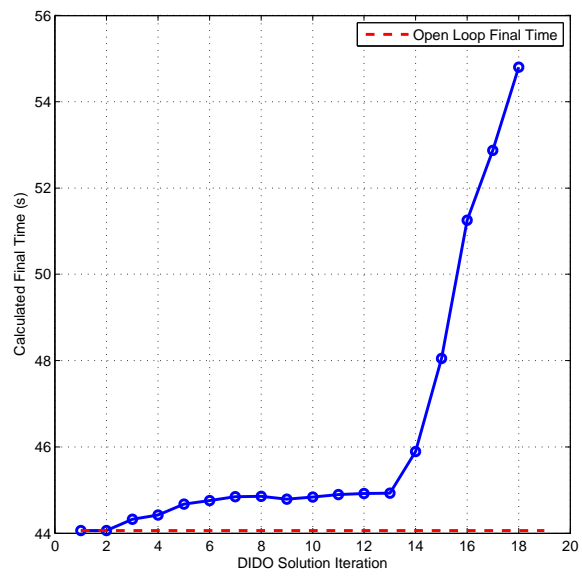


Figure 5.32: Maneuver Final Time - 10% MOI Variation for Real-time Update RTOC Controller 180° Maneuver

VI. Conclusions and Recommendations

6.1 Conclusions

This research was focused on developing, evaluating, and testing a near real-time optimal control (RTOC) spacecraft attitude controller for the AFIT's 2nd generation ground-based simulated satellite testbed, SimSAT II. It began by first simulating an open-loop optimal control (OLOC), eigenaxis rotation optimal control (EROC), and RTOC controllers model in MATLAB. The OLOC, EROC, and RTOC controllers used the pseudospectral-based optimal control solving program, DIDO, to solve the coupled and highly nonlinear equations of motion of SimSAT II, which cannot be solved analytically. After completing high fidelity OLOC, EROC, and RTOC controller modeling and simulations, the OLOC and EROC controllers were implemented on the SimSAT II testbed and experimentally tested. Prior to implementing an optimal controller on SimSAT II, several preliminary hardware integration and characterization steps were completed, including: hardware and software integration of a LN-200 Fiber-optic Gyroscope (FOG), characterization of SimSAT II's mass moment of inertia properties, and control torque characterization of SimSAT II's simulated thrusters.

RTOC has demonstrated the ability to provide significant performance improvement over current spacecraft control techniques for large angle agile slew maneuvers in the presence of parametric uncertainty in the spacecraft dynamic model. This research has shown through simulation that a RTOC controller can reduce the reorientation time for large angle spacecraft slew maneuvers by approximately 20%. The RTOC controller can also effectively perform large angle spacecraft slew maneuvers with a 10% uncertainty in the spacecraft's MOI matrix. However, before a RTOC controller is implemented on a satellite, further study is needed to better understand how RTOC controller parameters can be manipulated to achieve consistent and desirable performance.

This research also demonstrated that for a RTOC controller to function effectively, new optimal control solutions must be accurate and they must be updated rapidly enough to prevent phase lag induced controller error. The RTOC controller developed for this research only used DIDO to calculate optimal control solutions and therefore the performance of the RTOC controller was highly dependent on DIDO's solution accuracy and

computation time. Presently, DIDO runs in a MATLAB environment on a Windows-based PC. Since most satellites do not operate in this computing environment, the results shown in this thesis may not be completely indicative of the performance of a satellite using a RTOC controller. To validate the performance results found in this thesis, further simulation and experimental testing of the RTOC controller should be performed using satellite flight-compatible hardware and software.

Experimental testing of a RTOC controller using the SimSAT II simulated satellite was not conducted due to the time constraints for completing this research. However, an OLOC and EROC controller have both been implemented and experimentally tested on SimSAT II, effectively laying the groundwork for future RTOC controller implementation. Results of OLOC and EROC controller testing on SimSAT II suggest that significant parametric uncertainty still exist in the current estimation of SimSAT II's MOI matrix, the characterization of the thruster simulators, and the effects of air drag. Development and testing also suggest that there are several unmodeled sources of rotational error on SimSAT II including: the torque created by rotation of Maxon EC motor's axles, the air circulation of the SimSAT II laboratory, and the misalignment of SimSAT II's center of rotation and center of mass.

6.2 Recommendations for Future Research and Development

Below is a list of future research suggestions relating to RTOC and the further development of SimSAT II.

One of the errors in the current dynamics model for SimSAT II is that it does not account for the angular momentum of the Maxon EC motors in SimSAT II's EOMs. As the fans increase or decrease RPM, the change in angular momentum is large enough to cause a visible torque on the system. This torque on SimSAT II can be considerably reduced by spinning each set of fans in opposite directions so the total angular momentum for each set of fans is zero. However, SimSAT II uses a CAN network for distributing RPM commands to the fans, which sends commands serially around the network. This means that Fan #1 may receive its command and change its RPM setting before Fan #4 can do so, even though both RPM commands were sent simultaneously. As a result,

the total angular momentum between Fan #1 and Fan #4 is no longer zero which can result in an undesired torque on the system.

Currently SimSAT II's center of mass is not exactly aligned with the center of rotation of the air bearing. This causes the spacecraft to be in an unstable configuration that will tilt due to gravity causing a torque about the CR. This unstable configuration is very similar to an inverted pendulum tilting away from its unstable equilibrium point as seen in Figure 2.10. Boynton suggests counterbalancing the satellite or splitting the satellite into two halves to ensure CG is located at the CR of the air bearing [9]. Ross on the other hand, suggests actively compensating for the offset between the spacecraft CG and the air bearing center of rotation in the spacecraft's EOM [43]. Another option is to develop an automatic balancing tool that uses measured angular rate information from the LN-200 to actuate a moveable mass. No matter what method is used to resolve the "gravity torque," a new, higher fidelity measurement of SimSAT II's mass properties is needed.

Future research that will make use of SimSAT II's non-optimal PID controller, will need to develop new gain settings for the controller. Mathematical controller design methods were not used to find the gain values currently in SimSAT II's non-optimal PID controller since development of a non-optimal control was not necessary for this research effort. As a result, when using the non-optimal PID controller, SimSAT II can only rotate approximately 60° about the yaw axis before going unstable.

The results of the RTOC simulations performed for this research need to be validated by implementing a RTOC controller on SimSAT II's hardware and performing testing similar to the simulations. In order to implement a RTOC controller on SimSAT II, the communication structure for passing control and state information between the Ground Station PC and the SimSAT II's Mini-Box needs to be updated. Currently, SimSAT II uses the Remote Desktop Connection (RDC) program native to Windows XP to allow the transferring of files containing state measurements and control commands. Initial testing of an RTOC controller demonstrated that the time taken for RDC to read or write a file is inconsistent. This resulted in file reading and writing errors on both the Ground Station PC and the SimSAT II's Mini-Box PC which prevented the RTOC con-

troller from functioning correctly. One method for fixing the communication structure between the Ground Station PC and the SimSAT II's Mini-Box is to use User Datagram Protocol (UDP) to transfer files because UDP is structured for faster data transfer than the protocols being used by RDC. UDP communication between a ground station and a simulated satellite has been implemented effectively on the Naval Postgraduate School's simulated satellite, NPSAT1.

Bibliography

1. *Merriam-Webster's Collegiate Dictionary, Eleventh Edition*. Merriam-Webster Incorporated, 2003.
2. <http://en.wikipedia.org/wiki/butterworthfilter>. In *Butterworth Filter*, 2009.
3. <http://www.aerotech.com/products/airbearing/ahl9000.html>. In *AHL9000 Air Bearing Stage*, 2009.
4. John D. Anderson. *Fundamentals of Aerodynamics, 3rd Edition*. McGraw-Hill, 2001.
5. Jasbir S. Arora. *Introduction to Optimum Design*. McGraw-Hill Book Company, 1989.
6. Jr. Arthur E. Bryson. *Dynamic Optimization*. Addison-Wesley, 1999.
7. Nazareth Bedrossian, Sagar Bhatt, Mike Lammers, Louis Nguyen, and Yin Zhang. First Ever Flight Demonstration of Zero Propellant Maneuver™ Attitude Control Concept. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, August 2007.
8. Karl D. Bilimoria and Bong Wie. Time-Optimal Three-Axis Reorientation of a Rigid Spacecraft. *Journal of Guidance, Control, and Dynamics*, 16(3):446–452, 1993.
9. Richard Boynton. Using A Spherical Air Bearing to Simulate Weightlessness. In *55th Annual Conference of the Society of Allied Weight Engineers*, June 1996.
10. Richard Boynton. Mass Properties Measurement Handbook. In *Annual Conference of the Society of Allied Weight Engineers*, May 1998.
11. Richard Boynton. How Mass Properties Affect Spacecraft Attitude Control. In *67th Annual Conference of the Society of Allied Weight Engineers*, May 2008.
12. Richard Boynton and Kurt Wiener. How to Calculate Mass Properties (An Engineer's Practical Guide).
13. Glenn Research Center. <http://www.grc.nasa.gov/www/k-12/airplane/propth.html>. In *Propeller Thrust*, July 2008.
14. S. Cho, J. Shen, N. M. McClamroch, and D. S. Bernstein. Equation of Motion for the Triaxial Attitude Control Testbed. In *Proceedings of the Conference on Decision and Control*, December 2001.
15. H. Choset and D. Kortenkamp. Path Planning and Control for Free-Flying Inspection Robot in Space. *Journal of Aerospace Engineering*, 12:74–81, 1999.
16. J. E. Colebank, R. D. Jones, G. R. Nagy, R. D. Pollak, and D. R. Mannebach. SIMSAT: A Satellite Simulator and Experimental Test Bed for Air Force Research. In *Proceedings of the AIAA Space Technology Conference and Exposition*, September 1999.
17. T. Corazzini, A. Robertson, J. C Adams, A. Hassibi, and J. P. How. Experimental Demonstration of GPS as a Relative Sensor for Formation Flying Spacecraft. *Navigation: Journal of the Institute of Navigation*, 45(3):195–207, 1996.

18. N. G. Creamer, G. Kirby, R. Weber, A. Bosse, and S. Fisher. A Integrated GPS / Gyro / Smart Structures Architecture for Attitude Determination and Baseline Metrology. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 1998.
19. Capt Vincent J. Dabrowski. *Experimental Demonstration of an Algorithm to Detect the Presence of a Parasitic Satellite*. M.S. Thesis, Air Force Institute of Technology, 2003.
20. S. S. F. de Cordova and D. B. DeBra. Mass Center Estimation of a Drag-Free Satellite. In *Proceedings of the 6th Triennial World Congress of the International Federation of Automatic Control*, August 1975.
21. Robert W. Deters and Michael S. Selig. Static Testing of Micro Propellers. In *26th AIAA Applied Aerodynamics Conference*, August 2008.
22. Andrew Fleming, Pooya Sekhavat, and I. Michael Ross. Minimum-Time Reorientation of an Asymmetric Rigid Body. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, August 2008.
23. Bengt Fornberg. *A Practical Guide to Pseudospectral Methods*. Cambridge University Press, 1995.
24. H. Fornhoff. Air bearing platform, T50-2 Final report. *NASA Contractor Report NASA-CR-97588*, 1967.
25. Capt David B. French. *Hybrid Control Strategies for Rapid, Large Angle Satellite Slew Maneuvers*. M.S. Thesis, Air Force Institute of Technology, 2003.
26. Capt Jason W. Geitgey. *The Determination of Remaining Satellite Propellant Using Measured Moments of Inertia*. M.S. Thesis, Air Force Institute of Technology, 2004.
27. W. Haeussermann and H. Kennel. A Satellite Motion Simulator. *Astronautics*, 5:22–23, 90–91, 1960.
28. Martin Hepperle. <http://www.mh-aerotoools.de/airfoils/propuls3.htm>. In *Aerodynamic Characteristics of Propellers*, August 2003.
29. M. O. Hilstad. *A Multi-Vehicle Testbed and Interface Framework for the Development and Verification of Separated Spacecraft Control Algorithms*. M.S. Thesis, Massachusetts Institute of Technology, 2002.
30. 2Lt Michael S. Hines. *Fuel Estimation Using Dynamic Response*. M.S. Thesis, Air Force Institute of Technology, 2007.
31. David G. Hull. *Optimal Control Theory for Applications*. Springer, 2003.
32. Allan Y. Lee and Julie A. Wertz. In-Flight Estimation of the Cassini Spacecraft's Inertia Tensor. *Journal of Spacecraft*, 39(1):153–155, 2001.
33. Ronald Lee. <http://dx12.jsc.nasa.gov/about/whatisnb.shtml>. In *Sonny Carter Training Facility - Neutral Buoyancy Laboratory*, April 2007.
34. F. Li, M. Bainum, N. G Creamer, and S. Fisher. Rapid Reorientation Maneuvers of Experimental Spacecraft with a Pendulum Appendage. *Journal of Guidance, Control, and Dynamics*, 21:164–171, 1998.

35. S. Matunaga, K. Yoshihara, T. Takahashi, S. Tsurumi, and K. Ui. Ground Experiment System for Dual-Manipulator-Based Capture of Damaged Satellites. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2000.
36. Robert J. McGhee and Betty S. Walker. Performance Measurements of an Airfoil at Low Reynolds Numbers. In *Low Reynolds Number Aerodynamics Conference*, June 1989.
37. J. L. Meyer, W. B. Harrington, B. N. Agrawal, and G. Song. Application of Piezoceramics to Vibration Suppression of a Spacecraft Flexible Appendage. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, July 1996.
38. B. Pond and I. Sharf. Experimental Demonstrator of Flexible Manipulator Trajectory Optimization. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 1999.
39. Capt Neal Roach, Capt Wayne Rohe, and 1Lt Nate Welty. *A Systems Engineering Approach to The Design of a Spacecraft Dynamics and Control Testbed*. M.S. Thesis Defense, Air Force Institute of Technology, 2008.
40. Capt Neal R. Roach, Capt Wayne C. Rohe, and 1Lt Nathan F. Welty. *A Systems Engineering Approach to the Design of a Spacecraft Dynamics and Control Testbed*. M.S. Thesis, Air Force Institute of Technology, 2008.
41. I. Michael Ross. A Beginner's Guide to DIDO (Ver. 7.3), A MATLAB Application Package for Solving Optimal Control Problems, 2007.
42. I. Michael Ross, Qi Gong, and Pooya Sekhavat. The Bellman Pseudospectral Method. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, August 2008.
43. I. Michael Ross, Pooya Sekhavat, Andrew Fleming, and Qi Gong. Pseudospectral Feedback Control: Foundations, Examples and Experimental Results. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, August 2006.
44. Dominic Del Rosso. In *NASA Reduced Gravity Research Program*, March 2008.
45. H. Schubert and J. How. Space Construction: An Experimental Testbed to Develop Enabling Technologies. In *Proceedings of the Conference on Telemicrooperator and Telepresence Technologies IV*, October 1997.
46. Jana L. Schwartz, Mason A. Peck, and Christopher D. Hall. Historical Review of Spacecraft Simulators. *American Astronomical Society*, 2003.
47. J. Shen, N. H McClamroch, and A. M Bloch. Local Equilibrium Controllability of the Triaxial Attitude Control Testbed. In *Proceedings of the 41st IEEE Conference on Decision and Control*, December 2002.
48. Capt Jason E. Smith. *Attitude Model of a Reaction Wheel/Fixed Thruster Based Satellite Using Telemetry Data*. M.S. Thesis, Air Force Institute of Technology, 2005.
49. Y. Toda, T. Iwata, K. Machida, A. Otuka, H. Toriu, Y. Shinomiya, Y. Fukuda, M. Asakura, and N. Matuhira. Development of Free-Flying Space Telerobot, Ground

- Experiments on Two-Dimensional Flat Test Bed. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, August 1992.
50. James R. Wertz. *Spacecraft Attitude Determination and Control*. D. Reidel Publishing Company, 1985.
 51. James R. Wertz and Wiley J. Larson. *Space Mission Analysis and Design, 3rd Edition*. Microcosm Press, 2007.
 52. Bong Wie. *Spacecraft Vehicle Dynamics and Control*. AIAA Education Series, 1998.
 53. Bong Wie and Peter M. Barba. Quaternion Feedback for Spacecraft Large Angle Maneuvers. *Journal of Guidance, Control, and Dynamics*, 8(3):360–365, 1985.
 54. William E. Wiesel. *Spaceflight Dynamics, 2nd Edition*. McGraw-Hill, 1995.

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> OMB No. 0704-0188		
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 26-03-2007		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From — To) 25 Jun 07 – 25 Mar 09	
4. TITLE AND SUBTITLE Near Real-Time Closed-Loop Optimal Control Feedback for Spacecraft Attitude Maneuvers			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) C. Douglas McFarland, Capt, USAF			5d. PROJECT NUMBER ENY 09-142		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENY) 2950 Hobson Way WPAFB OH 45433-7765			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GA/ENY/09-M06		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Dr. Adam M. Fosbury Air Force Research Lab Space Vehicle Directorate Bldg 472, Rm 227A 3550 Aberdeen Drive Kirtland AFB, NM 87117			10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RVSV		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Optimization of spacecraft attitude maneuvers can significantly reduce attitude control system size and mass, and extend satellite end-of-life. Optimal control theory has been applied to solve a variety of open-loop optimal control problems for terrestrial, air, and space applications. However, general application of real-time optimal controllers on spacecraft for large slew maneuvers has been limited because open-loop control systems are inherently vulnerable to error and the computation necessary to solve for an optimized control solution is resource intensive. This research effort is focused on developing a near real-time optimal control (RTOC) system for spacecraft attitude maneuvers on the Air Force Institute of Technology's 2nd generation simulated satellite, SimSat II. To meet the end goal of developing a RTOC controller, necessary preliminary steps were completed to accurately characterize SimSAT II's mass properties and attitude control system. Using DIDO, a pseudospectral-based optimal control solver package, to continuously solve and execute a sequence of optimized open-loop control solutions in near real-time, the RTOC controller can optimally control the state of the satellite over the course of a large angle slew maneuver. In this research, simulation and experimental results clearly demonstrate the benefit of RTOC versus other non-optimal control methods for the same maneuver.					
15. SUBJECT TERMS satellite, control, optimal control, feedback control, thruster characterization, MOI, MOI characterization					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 149	19a. NAME OF RESPONSIBLE PERSON Eric Swenson, Lt Col, USAF (ENY)
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include Area Code) (937) 257-3636 x7479 eric.swenson@afit.edu