

Development and Validation of Guidance, Navigation, and Control Strategies for
CubeSat Earth Observation Missions

by

Elijah Vautour

Submitted in partial fulfilment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
November 2022

© Copyright by Elijah Vautour, 2022

Table of Contents

List of Tables.....	iv
List of Figures	v
Abstract.....	vii
List of Abbreviations and Symbols Used	viii
Acknowledgements	xiv
Chapter 1. Introduction.....	1
Chapter 2. Background.....	3
2.1 Orbital Mechanics	3
2.2 Global Positioning System (GPS)	5
2.3 Literature Review	6
Chapter 3. ODS Simulator Development	16
3.1 GRACE-FO Mission Data	17
3.2 Software Organization.....	17
3.3 GPS Sensor Model	18
3.4 RDEKF Algorithm	21
3.5 ODS Simulation Testing.....	25
Chapter 4. OGNS Development.....	35
4.1 Target Flyby Prediction	36
4.2 OGNS Algorithm Implementation	40
4.3 OGNS Simulation Testing	42
Chapter 5. ADCS Device Testing	45
5.1 RSS Platform.....	46
5.2 Reaction Wheel Testing.....	48
5.3 RSS Slew Controller Design.....	64

5.4	Sun Sensor Testing	73
Chapter 6.	Conclusions and Recommendations.....	91
6.1	ODS Simulator Testing Results	91
6.2	ADCS Device Testing Results	92
6.3	Contributions.....	93
6.4	Future Work	94
References.....		95
Appendix A.	Algorithms.....	104
Appendix B.	Flowcharts	109
Appendix C.	Additional Results.....	111
C.1	RDEKF Sensitivity Analysis Testing.....	111
C.2	OGNS Testing.....	113
C.3	ADCS Device Testing.....	119

List of Tables

Table 1. ODS technique results summary.....	33
Table 2. 96-hour OGNS testing results summary: part 1	43
Table 3. 96-hour OGNS testing results summary: part 2	43
Table 4. Select LORIS RWA mechanical and electrical properties.....	53
Table 5. RWA viscous friction motor model parameters	56
Table 6. Goodness-of-fit for the identified viscous friction motor models	56
Table 7. RWA viscous-Coulomb friction motor model parameters	59
Table 8. Goodness-of-fit for the identified viscous-Coulomb friction motor models.....	59
Table 9. RWA step input test summary: part 1.....	61
Table 10. RWA step input test summary: part 2	62
Table 11. RSS slew controller simulation tuning results summary: part 1.....	68
Table 12. RSS slew controller simulation tuning results summary: part 2.....	68
Table 13. RDEKF sensitivity analysis testing results summary: part 1	111
Table 14. RDEKF sensitivity analysis testing results: part 2.....	112
Table 15. 96-hour Halifax target flyby OGNS testing results summary	116
Table 16. 96-hour Rio target flyby OGNS testing results summary	117
Table 17. 96-hour Mumbai target flyby OGNS testing results summary.....	118
Table 18. Sun sensor intensity threshold sensitivity analysis results	123

List of Figures

Figure 1. SGP4 24-hour 3D RMSE	27
Figure 2. SGP4 14-day 3D RMSE	27
Figure 3. GPS SPS 24-hour 3D error	29
Figure 4. RDEKF initial condition (IC) experiment results	30
Figure 5. RDEKF model configuration experiment results	32
Figure 6. 2D Projection of satellite in Earth orbit illustrating target-satellite visibility criterion.....	36
Figure 7. GRACE-FO satellite ground tracks illustrating descending passes of the target latitude line for Halifax, Canada.....	38
Figure 8. GRACE-FO satellite ground track illustrating the longitudinal separation from the target viewing cone during a descending pass of the target latitude line.	39
Figure 9. Orbital guidance and navigation system process used in this research	41
Figure 10. OGNS 3D position error vs. time for the first Halifax target flyby on Day 226 of 2018	42
Figure 11. Robotic Spacecraft Simulator main subsystems/components.	46
Figure 12. RSS system overview	47
Figure 13. Reaction wheel system diagram	48
Figure 14. LORIS reaction wheel assembly used in this research.	52
Figure 15. RWA positive step input test summary plots for set point currents within the motors nominal operating range	54
Figure 16. Simulated response of the viscous friction motor model.....	57
Figure 17. Simulated response of the viscous and viscous-Coulomb friction motor models	60
Figure 18. RWA positive step input test summary plots for set point currents outside the motors nominal operating range	62
Figure 19. RWA constant positive acceleration speed response.....	63
Figure 20. RSS open-loop system for the RSS attitude control simulator	65
Figure 21. RSS closed-loop slew controller	65
Figure 22. 20-degree slew trajectory for testing the RSS slew controller	67
Figure 23. RSS slew controller experimental testbed.....	70

Figure 24. RSS slew controller performance summary for a 20-degree there-and-back slew trajectory test.....	71
Figure 25. RSS slew controller performance summary for a 30-degree there-and-back slew trajectory test.....	72
Figure 26. Photodiode light vector geometry.....	73
Figure 27. 3D Sun vector estimate parameters of Equation 5.21.....	75
Figure 28. AM0 solar power spectral distribution	77
Figure 29. 3D Printed version of the LORIS Sun sensor used on the RSS platform.....	79
Figure 30. RSS Sun sensor experimental testbed.....	80
Figure 31. RSS Sun sensor testing configuration.....	82
Figure 32. RSS Sun sensor experiment slew trajectory.....	83
Figure 33. Sun sensor mask intensities: raw vs. filtered.....	84
Figure 34. Sun sensor intensity threshold sensitivity analysis trends	86
Figure 35. Sun sensor state comparison	87
Figure 36. Sun sensor slew angle estimation summary	87
Figure 37. Sun sensor polar plots	90
Figure 38. Reduced dynamic extended Kalman filter for GPS-based onboard orbit determination.....	109
Figure 39. ODS simulator process flowchart.....	110
Figure 40. OGNS 3D position error vs. time for Halifax target flybys.....	113
Figure 41. OGNS 3D position error vs. time for Rio target flybys	114
Figure 42. OGNS 3D position error vs. time for Mumbai target flybys	115
Figure 43. RWA negative step input test summary plots for set point currents within the motors nominal operating range	119
Figure 44. RWA negative step input test summary plots for set point currents outside the motors nominal operating range	119
Figure 45. RWA constant negative acceleration speed response.....	120
Figure 46. RSS slew controller simulation tuning test 4 results summary	121
Figure 47. RSS slew controller simulation tuning test 5 results summary	121
Figure 48. 30-degree slew trajectory for testing the RSS slew controller	122

Abstract

This thesis describes the development of tools to design and test guidance, navigation, and control strategies for CubeSat Earth observation missions. The first contribution of this research is a novel orbital guidance and navigation system (OGNS) algorithm which minimizes onboard GPS usage while ensuring fine orbital accuracy during target flybys. Tests completed for three separate target locations show the OGNS algorithm successfully reduces GPS duty cycle to less than 1%. The second contribution of this research is a newly developed experimental platform used to test a custom reaction wheel and Sun sensors designed for the LORIS satellite. This testbed uses a planar air bearing platform as opposed to typical setups which use spherical air bearing platforms. A reaction wheel hardware model was identified and then validated using the testbed. Overall trends are observed for the Sun sensors; improvements to the setup will allow for absolute performance characterization.

List of Abbreviations and Symbols Used

Acronyms

ACS	attitude control system
ADC	attitude determination and control
ADCS	attitude determination and control system
ADR	active debris removal
ADS	attitude determination system
AM0	air mass zero
BF	body frame
CKF	cubature Kalman filter
COTS	commercial-off-the-shelf
DF	dual frequency (receivers)
DSSL	Dalhousie Space Systems Laboratory
ECEF	Earth-centered Earth-fixed
ECI	Earth centered inertial
EKF	extended Kalman filter
FOV	field-of-view
GNC	guidance, navigation, and control
GPS	global positioning system
GRACE	gravity recovery and climate experiment
IC	initial condition
ICD	interface control document
IMU	inertial measurement unit
ISL	Intelligent Systems Laboratory
LEO	low Earth orbit
LORIS	Low Earth Orbit Reconnaissance Imagery Satellite
MOCAP	motion capture system
NORAD	North American Aerospace Defense Command

OBC	onboard computer
ODS	orbit determination system
OGNS	orbital guidance and navigation system
POD	precise orbit determination
RDEKF	reduced dynamic extended Kalman filter
RMSE	root mean square error
ROS	robot operating system
RSS	Robotic Spacecraft Simulator
RWA	reaction wheel assembly
SF	single frequency (receivers)
SGP4	simplified general perturbations model 4
SPS	single point solution
TB	two-body force model
TBJ2	two-body force model which includes the J2 Earth oblateness perturbation
TLE	two-line element
TLL	target latitude line
U	unit equal to 10 cm^3
UKF	unscented Kalman filter

Symbols

\mathbf{r}	[m]	satellite position (ECEF coordinates unless specified otherwise)
$\dot{\mathbf{r}}$	[m s ⁻¹]	satellite velocity
$\ddot{\mathbf{r}}$	[m s ⁻²]	satellite acceleration
μ	[m ³ s ⁻²]	Earth's gravitational parameter
G	[m ³ kg ⁻¹ s ⁻²]	universal gravitational constant
m_{\oplus}	[kg]	Earth mass
\mathbf{p}	[m s ⁻²]	perturbative acceleration term
\mathbf{x}	-	state vector
$\dot{\mathbf{x}}$	-	state vector derivative
ρ_{s_i}	[m]	measured pseudorange/pseudorange observable for tracked GPS satellite i
b	-	receiver clock bias or viscous friction coefficient
c	[m s ⁻¹]	speed of light (3×10^8)
t_r	[s]	time of signal reception
t_s	[s]	time of signal transmission
D_{s_i}	[m]	true geometric range for tracked GPS satellite i
b	[s ⁻¹]	receiver clock bias
ϵ_{ρ_i}	[m]	pseudorange errors for tracked GPS satellite i
\mathbf{r}_{s_i}	[m]	position of tracked GPS satellite i (ECEF coordinates)
$\mathbf{R}_z(\theta)$	-	standard rotation matrix about z-axis by angle θ
ω_{\oplus}	[rad s ⁻¹]	Earth rotation rate
$\hat{\rho}_{s_i}$	[m]	predicted pseudorange for tracked GPS satellite i
$\hat{\mathbf{r}}$	[m]	predicted satellite position
\hat{b}	[s ⁻¹]	predicted receiver clock bias
$\hat{\mathbf{1}}_i^T$	-	line-of-sight unit vector for tracked GPS satellite i
$\Delta\rho$	[m]	pseudorange measurement residual vector

\mathbf{G}	-	measurement connection matrix
f	[s ⁻²]	receiver clock bias drift
$\hat{\mathbf{x}}_k^-$	-	a priori state estimate
\mathbf{p}_k^-	-	a priori estimate covariance
$\hat{\mathbf{x}}_k^+$	-	posteriori state estimate
\mathbf{p}_k^+	-	posteriori estimate covariance
\mathbf{x}_p	-	satellite state vector
\mathbf{x}_c	-	receiver clock state vector
t_{k-1}	[s]	previous time step
t_k	[s]	current time step
Δt	[s]	sample time
$\boldsymbol{\phi}$	-	state transition matrix
\mathbf{Q}	-	process noise covariance matrix
\mathbf{H}	-	observation matrix
\mathbf{K}	-	Kalman gain
\mathbf{R}	-	measurement noise covariance matrix
σ_r	[m ²]	position state variance
$\sigma_{\dot{r}}$	[m ² s ⁻²]	velocity state variance
s_b	-	1 st white noise spectral amplitude term [7]
S_f	-	2 nd white noise spectral amplitude term [7]
Δt_m	[s]	measurement time step
θ_T	[deg]	target visibility angle
R_\oplus	[m]	Earth radius
h	[m]	satellite altitude
λ_T	[deg]	target latitude
ϕ_T	[deg]	target longitude
t_{rise}	[s]	target flyby visibility rise time
t_{set}	[s]	target flyby visibility set time

\hat{P}_s	[s]	satellite mean orbital period
$\Delta\phi_v$	[deg]	longitudinal separation between target viewing cone and satellite
ϕ_s'	[deg]	satellite longitude after one full orbit
ϕ_s	[deg]	satellite longitude at current time step
λ_s'	[deg]	satellite latitude after one full orbit
λ_s	[deg]	satellite latitude at current time step
t	[s]	current time step
t'	[s]	time step after one full orbit
J	[kg m ⁻²]	flywheel inertia
$i(t)$	[A]	input motor current
$\dot{\theta}$	[rad s ⁻¹]	flywheel angular speed
$\ddot{\theta}$	[rad s ⁻²]	flywheel angular acceleration
τ_m	[N m]	motor torque
i_a	[A]	motor armature current
K_τ	[N m A ⁻¹]	motor torque constant
τ_f	[N m]	friction torque
b	[N m s]	viscous friction coefficient
c	[N m]	Coulomb friction coefficient
\boldsymbol{x}	-	motor model parameters
y_k	[rad s ⁻¹]	reaction wheel speed for current time step
y_{k-1}	[rad s ⁻¹]	reaction wheel speed for previous time step
u_{k-1}	[A]	applied motor current for previous time step
T_s	[s]	sample time
K_p	[mA deg ⁻¹]	slew controller proportional gain term
$\varepsilon(t)$	[deg]	slew error
$r(t)$	[deg]	reference slew angle
$\theta(t)$	[deg]	measured slew angle

T	[s]	slew maneuver duration
ω_{max}	[rad s ⁻¹]	RWA maximum speed range
I_{RW}	[kg m ⁻²]	RWA flywheel moment of inertia (spin axis)
L_{RW}	[N m s]	RWA momentum storage
θ_{slew}	[deg]	set point slew angle
α_{slew}	[deg s ⁻²]	set point slew acceleration
$\omega_{slew,max}$	[deg s ⁻¹]	maximum slew rate
$\alpha_{slew,max}$	[deg s ⁻²]	maximum slew acceleration
I_{RSS}	[kg m ⁻²]	RSS moment of inertia (spin axis)
τ_{max}	[N m]	RWA maximum continuous acceleration torque
I	-	measured photodiode intensity
I_0	-	light source intensity
η	-	photodiode measurement noise
\mathbf{n}	-	photodiode normal vector
\mathbf{s}	-	light source direction vector/Sun vector
\mathbf{H}	-	Sun sensor observation matrix
$\hat{\mathbf{s}}$	-	Sun vector estimate
$\hat{\theta}_s$	[deg]	Sun sensor slew angle estimate
\hat{s}_x	-	x-axis Sun vector component
\hat{s}_y	-	y-axis Sun vector component
α_s	[deg]	Sun sensor testing light source elevation angle
θ_s	[deg]	ground truth slew angle

Acknowledgements

To start, I would like to thank my supervisors Dr. Robert Bauer and Dr. Mae Seto for their support, direction, and expertise. Their guidance has helped me to grow substantially as an engineer and a researcher during my masters.

I would also like to acknowledge my colleagues in the Intelligent Systems Laboratory for their advice and friendship, both of which were very important factors towards the completion of this work.

Furthermore, I would like to thank my family and friends for their patience and support during the more challenging periods of this work, their shared enthusiasm and praise when things went well, and their overall willingness to listen to me talk about ‘space robots’ on more than a few occasions.

Finally, I would like to thank my supervisory committee members Dr. Kevin Plucknett and Dr. Serguei Iakovlev as well as the external contributors to this research, including the Natural Science & Engineering Research Council, the Canadian Space Agency, and the LORIS team for their hardware.

Chapter 1. Introduction

The Dalhousie University Space Systems Laboratory (DSSL) recently completed development of the Low Earth Orbit Reconnaissance Imagery Satellite (LORIS). LORIS is a 2U CubeSat with an onboard camera payload intended to capture images of the Nova Scotia coastline. Previous research [1] involved the development of an attitude determination and control system (ADCS) simulation tool, the ADCS simulator, used for the LORIS project. The ADCS simulator aims to accurately model the orbital and attitude dynamics which LORIS experiences in low Earth orbit (LEO). The intended use for the ADCS simulator is to design and validate onboard attitude determination and control (ADC) algorithms for LORIS and similar CubeSat missions in LEO.

This thesis describes the development of new tools to design, test, and validate guidance, navigation, and control (GNC) strategies for satellite Earth observation missions with limited onboard resources (the case with CubeSats). Research is directed towards two components of the ADCS simulator which require development: the 1. orbit propagator and 2. ADCS sensor and actuator models.

The ADCS simulator currently assumes perfect knowledge of the satellite's orbit. In reality, there are errors associated with any onboard orbit determination system (ODS). To capture the expected performance of the onboard ODS for LORIS and similar such missions, an ODS simulator is developed in this work which includes three common ODS navigation algorithms. Furthermore, the ODS simulator is used to develop a novel strategy to maximize accuracy during target flybys while minimizing the overall use of onboard resources.

Additionally, realistic sensor and actuator models are needed to capture the expected performance of both custom and commercial-off-the-shelf (COTS) devices used in the ADCS simulator. Experiments are therefore performed to identify accurate models which will improve simulation accuracy and characterize the performance of custom devices to evaluate their potential for use on future CubeSat missions. To this end, an experimental testbed is developed using the Dalhousie Intelligent Systems Laboratory's (ISL) robotic spacecraft simulator (RSS) platform. The RSS is a newly developed planar air bearing platform which emulates a two-dimensional microgravity environment on

Earth. Similar systems have traditionally been used to develop control strategies for LEO proximity operations [2,3,4,5]; in this research, the RSS is used for ADCS device testing.

In summary, the objectives of this thesis are to:

1. develop a realistic simulation environment to design and test ODS algorithms;
2. design an efficient onboard ODS strategy for CubeSat Earth observation missions;
3. develop an experimental testbed using the RSS platform to test ADCS devices and control strategies, and
4. evaluate the performance of custom equipment designed for the LORIS mission and identify models to improve the accuracy of the ADCS simulator

Furthermore, contributions from this research are:

1. an ODS simulator and RSS experimental testbed as tools to design, test, and validate GNC strategies for CubeSat Earth observation missions;
2. a novel strategy to maximize onboard orbit determination accuracy while minimizing the overall use of onboard resources (like GPS), and
3. a verified motor model for the LORIS reaction wheel assembly (RWA).

The following chapter provides context for the rest of the thesis, including background information on orbital mechanics, the global positioning system (GPS), and a literature review on the relevant topics studied here. In Chapter 3, the ODS simulator is described in detail in the implementation and testing of three commonly used navigation algorithms for satellite platforms in LEO. Chapter 4 describes the development of a combined strategy which can be used for accurate orbit determination during target flybys while minimizing use of onboard resources. Chapter 5 describes the ADCS device testing experiments conducted with the RSS platform. Finally, Chapter 6 provides a summary of the results, conclusions, and recommended future work.

Chapter 2. Background

This chapter begins with an introduction to orbital mechanics; concepts discussed in this subsection will be used to develop ODS strategies in later chapters. Section 2.2 provides a brief introduction to the GPS principles of operation. Finally, a literature review on topics related to CubeSats, orbit determination strategies, and ADCS device testing will be conducted in Section 2.3.

2.1 Orbital Mechanics

The fundamental equation for describing a satellite's trajectory in orbit is the two-body equation of relative motion. This equation, which is derived from Newton's law of gravitation, describes the acceleration of an orbiter with respect to a central body with an assumed point mass [6]. The two-body equation for an Earth orbiting satellite is shown in Equation 2.1:

$$\ddot{\mathbf{r}} = -\frac{\mu}{|\mathbf{r}^3|} \mathbf{r} \quad (2.1)$$

where,

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\mu = Gm_{\oplus} .$$

In Equation 2.1, \mathbf{r} is the 3D position of the satellite in an inertial frame of reference, $\ddot{\mathbf{r}}$ is acceleration, which is the second time derivative of position, and μ represents Earth's gravitational parameter which is equal to the universal gravitational constant G multiplied by the mass of Earth m_{\oplus} .

The orbit defined by the two-body equation is commonly referred to as a Keplerian orbit and can be described in terms of five orbital elements. These orbital elements define the shape and orientation of the orbit with respect to the central body. A sixth orbital element can locate the satellite's position on the orbit.

A satellite's instantaneous state in time may be described either in terms of its 3D position and velocity, or in terms of the six Keplerian orbital elements. Position and velocity may be used to derive the Keplerian orbital elements and vice versa. Furthermore,

if the satellite's state is known at one instant, the Keplerian orbital elements may be used to estimate its past or future states analytically [6].

2.1.1 Perturbed Orbits

Forces which cause a satellite's trajectory to deviate from a Keplerian orbit are referred to as perturbations. Perturbations in LEO are primarily a result of the non-spherical Earth, atmospheric drag, third body gravitational effects of the Sun and Moon, and solar radiation [6].

Equation 2.1 is modified to account for perturbation effects by adding a net perturbative acceleration term \mathbf{p} to the right-hand side of the equation, as shown in Equation 2.2:

$$\ddot{\mathbf{r}} = -\frac{\mu}{|\mathbf{r}^3|}\mathbf{r} + \mathbf{p}. \quad (2.2)$$

Equation 2.2 represents the modified equation of motion which describes the instantaneous acceleration of a satellite in a perturbed orbit. In general, the two-body acceleration is significantly larger in magnitude than that of the net perturbative acceleration term; however, the combined effects of perturbations will cause both secular and periodic variations of the Keplerian orbital elements.

2.1.2 Cowell's Method

Cowell's method is a classical approach to propagate a satellite's state forward in time. In this approach, the second order differential equations from Equation 2.2 are numerically integrated to predict the state vector \mathbf{x}

where,

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix}.$$

Taking the derivative of the state vector results in:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \end{bmatrix}$$

where $\ddot{\mathbf{r}}$ is calculated according to the sum of acceleration components acting on the satellite defined by Equation 2.2.

In this research, the fourth order Runge-Kutta method was used to integrate Equation 2.2 according to the formulation described in [6].

2.1.3 Coordinate Systems

As previously discussed, the equations of motion in Equations 2.1 and 2.2 are centered on an inertial reference frame. For Earth orbiting satellites, it becomes convenient to define an Earth centered inertial (ECI) coordinate system. The ECI coordinate system has its origin at Earth's center, its z-axis along Earth's rotational axis, and its x-axis pointed towards the vernal equinox. The ECI coordinate system is not truly inertial due to the acceleration of Earth with respect to the stars as well as the precession and nutation of Earth's pole [6]; however, for the applications considered in this research, it is considered to remain fixed.

While the ECI coordinate system defines a satellite's absolute position in space, it is often necessary to specify coordinates relative to the rotating Earth. For this we introduce the Earth-centered Earth-fixed (ECEF) coordinate system. The ECEF coordinate system has its origin at Earth's center, its z-axis aligned with Earth's rotational axis, and its x-axis pointed towards the prime meridian (line of longitude which is defined to be 0 degrees). The ECEF coordinate system is fixed with respect to Earth and therefore rotates with it. The ECI and ECEF coordinate systems are aligned whenever the prime meridian is in line with the vernal equinox.

Geodetic coordinates define the location of an object with respect to Earth's surface. Latitude and longitude specify angular coordinates with respect to Earth's equator and the prime meridian, respectively. Then, height specifies the orthogonal distance of an object from Earth's surface.

Finally, a coordinate system which is in the satellite's body frame (BF) relates a satellite's attitude to external reference frames.

2.2 Global Positioning System (GPS)

The GPS is a satellite-based radio navigation system developed and operated by the United States government. Its principle of operation is based on one-way ranging signals broadcast from the GPS satellite constellation [7]. For the duration of this section, the term satellite means an arbitrary satellite in the GPS satellite constellation which is transmitting information and the term receiver means the GPS receiver which tracks and receives information from the GPS satellite constellation.

The concept of ‘pseudorange’ is important to how GPS positioning works. Each satellite broadcasts navigational data which contains the information needed to solve for its ECEF position at the instant of transmission t_s . The time of signal reception t_r is recorded by the receiver and used to calculate the pseudorange observable, as shown in Equation 2.3 [8]:

$$\rho_{s_i} = c(t_r - t_s) . \quad (2.3)$$

In Equation 2.3, the measured pseudorange ρ_{s_i} for tracked satellite i is equal to the signal transmission time $t_r - t_s$ multiplied by the speed of light c . Assuming both the satellite’s onboard clock and the receiver’s clock are synchronized, and the signal is not delayed by external factors, the measured pseudorange will be equal to the true geometric range between receiver and satellite in an inertial coordinate frame.

In reality, both clocks exhibit some bias with respect to the true system time. The navigation data broadcast by the satellite includes information to correct the satellite transmission time; however, the receiver clock bias is an unknown state which must be estimated and corrected for. Furthermore, there are additional bias errors, propagation errors, and random errors which are present in the measured pseudorange that may need to be captured depending on the accuracy requirements.

In addition to the pseudorange observable, the GPS receiver may also track the Doppler shift and carrier-phase observables for improved performance. Furthermore, GPS signals transmit at multiple frequencies which may be used independently or together as part of the GPS observation model [7].

2.3 Literature Review

This section begins with a literature review on CubeSats to provide context for the recent increased relevance of these small satellite platforms, the advantages and disadvantages associated with their use, and areas for improvement with regards to improving mission success. Following that, research into the use of GPS in LEO, common orbit determination algorithms, and onboard ODS strategies will be discussed in Sections 2.3.2 – 2.3.4, respectively. Then, a literature review of common practices for ADCS device testing is provided in Section 2.3.5. The literature review will then be summarized in Section 2.3.6.

2.3.1 CubeSats

Satellites which weigh under 600 kg are considered small satellites and can be further categorized into mini, micro, nano, pico, and femto mass categories [9]. Features of small satellite platforms include lower cost and shorter development times at the expense of limited onboard resources. In 2013, small satellites represented less than 2% of the mass in Earth orbit; since then, small satellites have become the primary source to space access for commercial, government, private, and academic institutions worldwide [9].

This research focuses on a particular class of small satellite, CubeSats, which are further categorized in terms of their form factor. CubeSats are composed of standard 10 cm square cubes, or units (U), which weigh up to 1.33 kg each. CubeSats were originally conceived as small platforms for academic research in a collaboration between California Polytechnic State University and Stanford University in 1999 [10]. By 2018, there were over 855 CubeSat launches from countries around the world [11]. Furthermore, the scope of CubeSat missions has rapidly expanded to include high-quality scientific experiments at a fraction of the cost compared to traditional satellite platforms [10].

The low-cost and short development time of CubeSats make them an accessible platform to remove the daunting barrier-to-entry for space missions; this is particularly true for universities, some research institutions, and small-scale industry developers. However, studies suggest that many CubeSat projects still fail due to poor planning in the developmental phase [12,13]. In [13], a methodical approach for student-driven CubeSat projects is presented with emphasis given to proper planning and documentation. In [14], the integration and verification approach used during development of the ISTSat-1 CubeSat is presented; the authors suggest that iterative functionality testing is important to identify system-level errors early-on. The development of tools like the ADCS simulator in [1], and those in this research, aim to expand the capabilities of the DSSL to develop CubeSats that conduct quality scientific experiments with a high chance of mission success.

In the following subsection, GPS use in LEO and the availability of COTS GPS receivers for small satellite platforms is discussed.

2.3.2 GPS in LEO

The use of GPS in LEO has shown exceptional performance with respect to accurate satellite positioning [15,16]. The first spaceborne GPS receiver was operated on Landsat-4 in 1982 [17]. Following the achievements of Landsat-4, there have been many follow-up missions which have successfully flown high-grade custom geodetic receivers [18,19,20,21,22].

GPS is now considered the primary sensor to perform orbit determination in LEO [9]. In recent years, COTS GPS receivers have become more readily available for space applications; however, many of these are still unsuitable for platforms with extremely limited onboard resources [23]. The development of low-cost receivers, and other strategies utilizing GPS, continues to be an active area of research [23,24,25]. This research investigates onboard ODS' which efficiently use GPS to ensure suitability for CubeSat platforms such as LORIS.

For applications with limited onboard resources, the GPS receiver can be turned on as needed or duty cycled at regular intervals to conserve power; however, space operated GPS receivers also exhibit large startup delays which slows down their time-to-first-fix (TTFF). The TTFF depends on the startup characteristics of the receiver and are categorized as cold- or warm-starts [26]. Cold-starts occur when the GPS receiver has no prior knowledge of the GPS satellite constellation and can result in a TTFF that is greater than 20 minutes [25,26]. Warm-starts occur when the GPS receiver has almanac data for the GPS satellite constellation available (valid for a few weeks) and typically result in a TTFF of a few minutes [16,26,27]. For the Earth observation missions considered in this research, the GPS will be used at least once per day during target flybys; it is therefore expected that only warm-start delays will occur during regular mission operations. Therefore, a 3-minute TTFF was assumed acceptable for the ODS simulator tests completed in Chapters 3 and 4.

Moving forward, common techniques for satellite orbit determination are discussed.

2.3.3 Orbit Determination

Orbit determination is the process to estimate the orbital trajectory of an object in space from a set of observations or measurements. Classical orbit determination techniques were initially developed to track celestial bodies and have been around for centuries [28]. Over the last few decades, there has been significant research regarding the development of GPS-based precise orbit determination (POD) techniques to track satellites in LEO [29]. Most modern techniques can be broadly categorized as dynamic, kinematic, or reduced dynamic.

Dynamic orbit determination techniques use analytical or numerical force models to propagate a set of initial conditions forward (or backward) in time. The SGP4 algorithm is commonly used to estimate a satellite's orbit analytically from mean orbital elements which are supplied as two-line element sets (TLE). TLEs are available for most Earth orbiters from the North American Aerospace Defense Command (NORAD). NORAD TLEs are generally considered accurate to within a few kilometers, with position state estimation accuracy degrading by up to 1 – 3 kilometers per day from the TLE epoch [30]. Methods to improve the accuracy and longevity of NORAD TLEs with intermittent GPS measurements are investigated in [31,32,33]; however, these approaches still present relatively coarse estimates compared to other techniques.

Kinematic orbit determination techniques are based on the solutions to observation models which estimate a satellite's instantaneous state from a set of measurements. The GPS single point solution (SPS) is the least-squares solution to the pseudorange observation model and is a classical kinematic approach to GPS positioning [7]. The GPS SPS is sometimes referred to as the GPS navigation solution due to its frequent implementation on both single frequency (SF) and dual frequency (DF) GPS receivers [34]. When applied to the orbit determination problem, the GPS SPS has shown positional accuracy on the order of 8 – 10 m with appropriate data editing [35]. Furthermore, accuracies on the order of 1 – 1.5 m can be achieved with alternative observation models which combine pseudorange and carrier-phase measurements [36]. Estimates are only generated when the receiver is on and at least 4 GPS satellites are being tracked which make kinematic techniques like GPS SPS susceptible to instances of poor performance due to poor observation geometry [7].

Reduced dynamic techniques combine knowledge from GPS measurements and numerically integrated force models to better estimate a satellite's state; this solution attempts to correct both the estimation drift which occurs with dynamic propagators and the gaps which occurs with purely kinematic estimation techniques [37]. A common implementation of the reduced dynamic technique is based on the extended Kalman filter (EKF) formulation [38]. Both real-time and post-processed solutions were considered for GPS measurements from SF and DF receivers with variable GPS duty cycles. It was shown that real-time positional estimation accuracies on the order of 1 m were maintained with a 50% GPS duty cycle using DF receiver measurements; a similar test conducted using SF measurements resulted in accuracies on the order of 10 m. The accuracy and robustness of the reduced dynamic technique make it one of the most commonly implemented solutions for POD.

There are studies which evaluate the impacts of the selected observation model, measurement duty cycle, and force model complexity on the reduced dynamic technique [38,39,40,41,42]; a brief summary of common trends from these studies is provided here. DF GPS receivers were generally found to produce more accurate results than SF receivers at the expense of increased onboard power consumption. SF receivers can yield comparable performance to DF receivers if both pseudorange and carrier-phase measurements are available; however, this often necessitates sophisticated GPS observation models. Finally, accurate force models may be used to minimize the GPS duty cycle, but this leads to increased computational complexity. In this research, only SF GPS pseudorange measurements and force models of minimal complexity are considered to ensure suitability for satellite platforms with limited onboard resources.

The EKF was the most commonly used state estimator in the literature; however, there are several variants of the EKF which have been tested with different degrees of success. In [43], the performance of the cubature Kalman filter (CKF) and the unscented Kalman filter (UKF) were compared to that of the EKF for a highly elliptical Molniya orbit. The purpose for their study was to illustrate the benefits of the CKF and UKF as alternative solutions to highly nonlinear problems in the absence of a continuous GPS fix; however, all three filters showed similar performance. The authors suggest this was because in the low altitude regions where the system dynamics were most nonlinear, GPS coverage was

excellent; alternatively, when the GPS struggled to maintain a fix, the system dynamics were relatively linear. They concluded the EKF was optimal for their application; however, the CKF and UKF may outperform the EKF for extremely low GPS duty cycles. In [44], several configurations of Schmidt-Kalman filter (SKF) are tested in real-time using flight data from the SJ-9A satellite mission and compared to a baseline test with the standard (linear) Kalman filter. The best performing SKF configuration resulted in a 3D position accuracy of 46.0 cm which was 6.4 cm better than the baseline test. These results suggest that alternative Kalman filter formulations may result in improved performance for missions with stringent accuracy requirements; however, the EKF can generally be used for most POD applications.

The required force model depends on a satellite's orbit. The effects of Earth's oblateness and atmospheric drag are more significant at low altitudes and the effects of solar radiation pressure and third-body gravity from the Sun and Moon are more significant at higher altitudes. Earth oblateness is by far the largest perturbation in LEO; this perturbation results in short-term periodic variations of a satellite's orbital trajectory [6]. Gravitational potential models account for Earth oblateness where higher order models are more accurate at the expense of increased computational complexity [34]. Atmospheric drag causes long-term secular variations of a satellite's orbital trajectory which can be significant at low altitudes (< 1000 km) [6]; however, drag deceleration is still relatively small in magnitude compared to Earth oblateness [34] and is therefore sometimes excluded to reduce model complexity while still maintaining accuracies on the order of tens of meters [45,46]. In this research, only the most significant perturbation effects of Earth oblateness are considered based on the J2 zonal harmonic terms [6].

The selected orbit determination technique for any satellite mission will depend on the application, requirements, and availability of onboard resources. For applications with limited onboard resources, the SGP4 algorithm using NORAD TLEs may provide sufficient estimation accuracy. If the onboard GPS receiver is always powered on, the GPS receivers navigation solution will provide superior estimation accuracy to the SGP4 algorithm; alternatively, a custom GPS SPS algorithm implementation can be efficiently used with raw pseudorange measurements. The reduced dynamic extended Kalman filter (RDEKF) algorithm accurately estimates the satellite's state with limited GPS availability;

however, its performance depends on the selected GPS receiver, measurement duty cycle, and observation and force model complexity. Furthermore, all three of these techniques may be used together or separately as needed by an onboard ODS. To be flexible to mission scenarios and increase the scope of the DSSL's capabilities, the SGP4, GPS SPS, and RDEKF algorithms were all implemented in the ODS simulator, as described in Chapter 3.

The following subsection introduces several onboard ODS strategies which have been successfully used on previous satellite missions.

2.3.4 Onboard Orbit Determination Systems

An onboard ODS provides real-time state estimation to improve ADCS performance. For example, the PROBA mini-satellite was developed by the European space agency in the early 2000s for an Earth observation mission with the objective to demonstrate the feasibility and benefits of onboard satellite autonomy [47]. The PROBA spacecraft GNC system included features for real-time orbit and attitude determination, prediction of Earth target flybys, and the planning and scheduling of spacecraft attitude operations; these features allowed for better management of PROBA's onboard resources and the reduction of ground-station operating costs. Similar examples of onboard autonomy are implemented on the BIRD [48], KOMPSAT-2 [49], PROBA-2 [50], and IUSTSAT [51] satellite missions.

The sophistication of the satellite's autonomy and the accuracy requirements for real-time positioning depends on the specific mission objective and availability of onboard resources. The PROBA satellite mission had a coarse accuracy requirement of 1 km with a goal of 200 m for medium-ranged terrestrial pointing [47]. The BIRD satellite mission had stricter requirements for a position accuracy of 90 m; however, in the case that the primary algorithm position variance exceeded a certain threshold, the SPG4 algorithm was used as a backup for coarse orbit determination [48]. The KOMPSAT-2 satellite mission split orbit determination into two software functions: the first was for onboard navigation purposes and the second used POD techniques for post-processing payload data with GPS measurements [49]. The IUSTSAT satellite mission used an EKF-based onboard ODS which simultaneously provided autonomous fault detection features and switched to the SGP4 algorithm as necessary [50].

For Earth observation missions like LORIS, accuracy requirements for the onboard ODS are largely dictated by the required ADCS performance during target flybys. In this research, the PROBA and BIRD satellite missions are used as baselines for defining coarse and fine positional accuracy requirements for onboard orbit determination. Accuracies on the order of a few kilometers (< 3 km) are considered for coarse orbit determination. Accuracies below 200 m are considered for fine orbit determination.

The GPS SPS and RDEKF algorithms are both likely to provide sufficient accuracy to the ADCS while pointing the camera payload. If onboard resources are limited, the RDEKF algorithm may be used with a reduced GPS duty cycle; however, the ADCS performance will be impacted by the resulting loss of accuracy. In this research, an alternative approach is investigated where an orbital guidance and navigation system (OGNS) is developed in Chapter 4 to automatically switch between coarse and fine ODS algorithms during target flybys.

The second thesis objective was to consider common practices for testing ADCS sensors and actuators with a specific focus on devices used on LORIS. The literature review for this is described next.

2.3.5 ADCS Device Testing

The LORIS ADCS simulator [1] currently uses idealized sensor and actuator models for devices that are common to CubeSat platforms, including magnetometers and magnetorquers, Sun sensors, gyroscopes, and reaction wheels. This section provides a list of experiments from the literature to create accurate models for these devices to evaluate their expected performance in simulation prior to implementation.

Magnetometers and magnetorquers measure and interact with Earth's magnetic field, respectively. Tests with these devices typically rely on a Helmholtz cage which reproduces the magnetic field conditions expected in LEO. In [52], a detailed description of the development and validation of a Helmholtz cage testing setup is given. Verification and calibration experiments for magnetometers and magnetorquers using a Helmholtz cage have been reported [53,54].

Sun sensors estimate the Sun vector with respect to the satellite BF and are typically tested using real or artificial sunlight conditions. In [55], the authors complete several experiments to verify the functionality, and characterize the performance, of a custom fine

Sun sensor using both direct sunlight and an array of white LEDs. In [56], custom coarse Sun sensors are tested using an artificial light source and evaluated based on their in-orbit performance on the UPMSat-2 satellite.

Gyroscopes measure a satellite's attitude rates and are typically integrated as part of an inertial measurement unit (IMU). In [57], the Allan variance method is used to characterize the noise characteristics of a gyroscope from a COTS IMU. Similar experimental analysis is completed in [58] for a COTS gyroscope which is used as part of the onboard navigation system on an unmanned aerial vehicle.

Reaction wheels are motor actuated flywheels which apply torques to a satellite to control its attitude. Reaction wheel testing typically involves characterizing the actuators' input-output dynamics. In [59], the design, integration, and verification of the Delfi-n3Xt reaction wheel system is described; their experiments included voltage step response tests where theoretical model parameters were manually tuned to match experimental results. [60] and [61] provide in-depth descriptions of experimental methods to estimate reaction wheel motor models for improved attitude control; their controller is based on the reaction wheel speed response from current inputs. In this research, current step input tests are used to identify motor model parameters similar to [60] and [61].

In addition to the verification, calibration, and system identification experiments of individual ADCS sensors and actuators, experiments which test these devices together verify the combined functionality of an onboard ADCS. A dynamic testbed for nanosatellite ADCS verification is presented in [62]; this testbed includes a spherical air bearing platform, Helmholtz cage, vision system, and Sun simulator. Similar testbeds which use some or all of the same features as [62] can be found in [53,63,64,65,66,67]. This research aims to complete similar ADCS experiments using a planar air bearing setup.

The scope of this research does not include development of a Helmholtz cage and therefore magnetometer and magnetorquers are not tested. Furthermore, particular focus was given to custom ADCS devices developed for LORIS and a novel reaction wheel assembly (RWA) and custom Sun sensors were therefore selected for testing. These devices were integrated onto the RSS platform in Chapter 5 to validate the testbed for ADCS experiments and conduct preliminary system identification and verification tests.

In the following subsection, the literature reviews provided in Sections 2.3.1 – 2.3.5 are summarized as they impact decisions made in the rest of this thesis.

2.3.6 Summary

The scope of CubeSat missions is rapidly expanding; however, to improve the likelihood of a successful mission, proper planning, documentation, and testing in the early phases of development is important. The tools developed in this research aim to provide a means to rapidly prototype, test, and validate GNC strategies to enable mission success.

An onboard ODS provides real-time state estimation to improve ADCS performance. GPS is the primary sensor to perform onboard orbit determination in LEO; however, use of the GPS on CubeSat platforms is limited due to the finite onboard power supply. The most common ODS strategies include the SGP4 algorithm, the GPS SPS (a.k.a. the GPS navigation solution), and the reduced dynamic extended Kalman filter (RDEKF); however, the use of these navigation algorithms will depend on specific mission requirements. All three algorithms are considered in this research to account for different mission scenarios; they are implemented and tested in a custom ODS simulator in Chapter 3.

The SGP4, GPS SPS, and RDEKF algorithms may be used independently, or fused, to meet specific mission requirements. Furthermore, the implementation of onboard autonomy can efficiently manage resources and improve fault tolerance in an onboard ODS. Chapter 4 introduces a novel approach to combine ODS strategies to minimize the use of onboard resources.

ADCS sensors and actuators commonly include magnetometers and magnetorquers, Sun sensors, gyroscopes, and reaction wheels; these devices are tested independently to identify sensor models to verify functionality, and evaluate performance. Furthermore, ADCS verification is often performed with some combination of Helmholtz cage, spherical air bearing platform, Sun lamp, and vision system. The RSS is a planar air bearing platform and is investigated in this research as an alternative approach to ADCS verification testing. In Chapter 5, a novel reaction wheel assembly and custom Sun sensors from LORIS are integrated onto the RSS platform and tested experimentally.

Chapter 3. ODS Simulator Development

The ODS simulator was developed to design and test onboard ODS strategies. Rather than simulating the space environment itself, the ODS simulator reproduces the navigation conditions which an onboard ODS would be expected to operate in. In this research, these conditions are largely dictated by the satellite's orbital trajectory and the availability of real-time onboard GPS telemetry. The goals for the ODS simulator development were to 1. reproduce real-time GPS measurements from a satellite mission in LEO, 2. implement and test navigation algorithms commonly used for onboard orbit determination, and 3. make recommendations for an ODS strategy which is suitable for CubeSat Earth observation missions with limited onboard resources.

Three commonly used navigation algorithms were considered and implemented as part of this research: SGP4, GPS SPS, and RDEKF. Advantages and disadvantages of these algorithms were discussed in Section 2.3.3 from the literature; they may be used separately, or fused together, to achieve specific mission requirements. All three algorithms were implemented in the ODS simulator to design strategies for different mission scenarios; however, tests completed in this research focus particularly on their suitability for CubeSat Earth observation missions.

Using GPS in LEO introduces challenges which can restrict the availability of continuous measurement updates. Furthermore, the dynamics of a satellite in orbit may significantly impact the quality of the GPS measurement. To accurately reproduce these conditions, a GPS sensor model was implemented in the ODS simulator in Section 3.3 and uses measurements from the GRACE-FO satellite mission [68].

The *satellitescenario()* function from MATLAB's Aerospace Toolbox generates the SGP4 navigation solutions used in the ODS simulator [69]. Custom implementations of the GPS SPS and RDEKF algorithms were developed and integrated in the ODS simulator in Sections 3.3.2 and 3.4, respectively. In Section 3.5, each of these algorithms are tested and evaluated based on their agreement with ground truth ephemeris from the GRACE-FO satellite mission; a combined ODS strategy is then developed in Chapter 4 based on the simulation results.

3.1 GRACE-FO Mission Data

Tests completed in this research are based on experimental results from the NASA GRACE-FO satellite mission for days 226 – 239 of 2018. The GRACE-FO mission is a successor to the gravity recovery and climate experiment (GRACE) mission which had the same goal of accurately mapping variations in Earth’s gravity. Two satellites were used in the GRACE-FO mission; however, all data used in this research is for satellite ‘A’. Additional details of the experiment including information regarding the satellite hardware, orbit shape, and available online data products can be found in [68].

Raw GPS pseudorange and ground truth satellite ephemeris measurements were retrieved from the JPL PO.DAAC database at [70]. Broadcast GPS navigation data used by the simulated GPS receiver were retrieved from the CDDIS database at [71]. Finally, TLE datasets used by SGP4 algorithm were retrieved from [72].

3.2 Software Organization

The ODS simulator was developed and implemented in the MathWorks MATLAB programming platform using a class-based software structure. Functions and variables were organized into classes based on their role in the simulator. Objects were then generated as class instances to simulate components of an onboard ODS. Class definitions used in the simulator include: GPS, EKF, NAV, PARAM, DATA, and PARSE.

The primary benefit of the ODS simulator is in its configurability. The GPS class can create a simulated GPS sensor object which generates GPS observables and SPS state estimates. The EKF class contains the functions used in the RDEKF algorithm and can generate a configurable EKF filter object which stores user-defined filter properties. The NAV class contains the actual navigation algorithm definitions which were tested as part of this research. The PARAM class contains the list of constant parameters used in the simulator including natural constants and mathematical conversions. The DATA class contains functions used for communication and data handling in the ODSS simulation environment. Finally, the PARSE class is used to parse the raw GRACE-FO datasets into standard MATLAB data structures for testing.

3.3 GPS Sensor Model

A pseudorange based GPS observation model was implemented in the ODS simulator as in Equation 3.1 [7]:

$$\rho_{s_i} = D_{s_i} + cb + \epsilon_{\rho_i} \quad (3.1)$$

where,

$$D_{s_i} = |\mathbf{r}_{s_i} - \mathbf{r}|.$$

In Equation 3.1, the measured pseudorange ρ_{s_i} between receiver and GPS satellite i is equal to the true geometric range D_{s_i} plus the receiver clock bias b multiplied by the speed of light constant. Additional sources of error which include the satellite clock bias, atmospheric delays, and relativistic effects are shown as ϵ_{ρ_i} .

The unknown states in Equation 3.1 are the receiver position \mathbf{r} and clock bias. These states can be estimated provided at least four satellites are near-simultaneously tracked. The GPS receiver decodes navigation messages from each of the tracked satellites which contain the necessary data to generate the pseudorange observable and satellite position \mathbf{r}_{s_i} .

In Section 3.3.1, the algorithm to calculate the GPS observables is described. This algorithm is the basis to convert raw GRACE-FO GPS measurements into observables which are used by the GPS SPS and RDEKF navigation algorithms. In Section 3.3.2, the GPS SPS algorithm is described.

3.3.1 GPS Observables

GPS navigation data used in the ODS simulator was limited to measurements transmitted on the C/A code of the L1 frequency signal. Raw pseudorange measurements tracked by a multi-channel receiver are assumed timestamped to the signal reception time t_r . In addition to the timestamped pseudorange measurements, the broadcast navigation data is provided for each tracked GPS satellite.

Section 20.3.3.3.1 [73] of the GPS interface control document (ICD) describes the procedure to estimate each tracked GPS satellite's clock bias b_{s_i} (a.k.a. the code phase time offset in the GPS ICD). The GPS satellite clock bias is used to correct the estimated signal transmission time and raw pseudorange measurement. Afterwards, the ECEF position at

signal transmission time is calculated for each tracked GPS satellite according to the procedure described in Section 20.3.3.4.3 of the GPS ICD.

Each of the tracked satellite positions must be transformed into a common reference frame. This is necessary due to the rotation of the ECEF frame during the finite interval during which each signal travelled from the transmitter onboard the GPS satellite to the GPS receiver [74]. Using Equation 3.2, the position of each GPS satellite may be transformed into common ECEF coordinates at the signal reception time t_r [75]. Note that Equation 3.2 represents a passive rotation of the satellite coordinate axes which means the physical position vectors remain unchanged:

$$\mathbf{r}_{s_i}' = \mathbf{R}_z(\theta) \mathbf{r}_{s_i} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{r}_{s_i} \quad (3.2)$$

where,

$$\theta = \omega_{\oplus} \frac{\rho_{s_i}}{c}.$$

Algorithm 3.1 in Appendix A summarizes the procedure described in this subsection. The outputs from Algorithm 3.1 include the $n \times 1$ vector of pseudorange observables $\boldsymbol{\rho}$ and the $n \times 3$ matrix of GPS satellite positions \mathbf{r}_s in ECEF coordinates at signal reception time t_r , where n represents the number of tracked satellites for each measurement update.

3.3.2 GPS SPS Algorithm

The GPS SPS is the least squares solution to the pseudorange measurement in Equation 3.3 [7]:

$$\hat{\rho}_{s_i} = |\mathbf{r}_{s_i} - \hat{\mathbf{r}}| + c\hat{b} \quad (3.3)$$

$$\Delta\rho_{s_i} = \hat{\rho}_{s_i} - \rho_{s_i} = [-\hat{\mathbf{1}}_i^T \quad 1] \begin{bmatrix} \Delta\mathbf{r} \\ c\Delta b \end{bmatrix} \quad (3.4)$$

where,

$$\hat{\mathbf{1}}_i^T = \frac{\mathbf{r}_{s_i} - \hat{\mathbf{r}}}{|\mathbf{r}_{s_i} - \hat{\mathbf{r}}|}$$

$$\Delta\mathbf{r} = \hat{\mathbf{r}} - \mathbf{r}$$

$$\Delta b = \hat{b} - b.$$

Starting with an *a priori* estimate of the receiver's position $\hat{\mathbf{r}}$ and clock bias \hat{b} , Equation 3.3 and Equation 3.4 relate the pseudorange measurement residual $\Delta\rho_{s_i}$ to the position state estimate error $\Delta\mathbf{r}$ and clock bias state estimate error Δb .

In Equation 3.4, $\hat{\rho}_{s_i}$ represents the predicted pseudorange calculated using Equation 3.3 and ρ_{s_i} represents the corrected pseudorange observable from Algorithm 3.1. For each satellite tracked by the receiver, the pseudorange measurement residual and estimated line-of-sight unit vector $\hat{\mathbf{1}}_i^T$ are calculated and combined to form Equation 3.5:

$$\Delta\boldsymbol{\rho} = \mathbf{G}\Delta\boldsymbol{x} \quad (3.5)$$

where,

$$\Delta\boldsymbol{\rho} = \begin{bmatrix} \Delta\rho_{s_1} \\ \Delta\rho_{s_2} \\ \vdots \\ \Delta\rho_{s_n} \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} -\hat{\mathbf{1}}_1^T & 1 \\ -\hat{\mathbf{1}}_2^T & 1 \\ \vdots & \vdots \\ -\hat{\mathbf{1}}_n^T & 1 \end{bmatrix} \quad \Delta\boldsymbol{x} = \begin{bmatrix} \Delta\mathbf{r} \\ c\Delta b \end{bmatrix}.$$

In Equation 3.5, \mathbf{G} represents the measurement connection matrix which relates the $n \times 1$ pseudorange measurement residual vector $\Delta\boldsymbol{\rho}$ to the 4×1 state estimate error vector $\Delta\boldsymbol{x}$. Equation 3.5 is rearranged to produce Equation 3.6 which is the least squares solution to the pseudorange measurement equation:

$$\Delta\hat{\boldsymbol{x}} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \Delta\boldsymbol{\rho}. \quad (3.6)$$

Equation 3.6 is repeated iteratively until the state estimate error vector is within a specified tolerance. A tolerance of 1×10^{-3} m was used to check the three-dimensional position range error $|\Delta\hat{\mathbf{r}}|$.

The GPS SPS only provides state estimates for the receiver position and clock bias. For this research, tracked states also include the receiver velocity $\dot{\mathbf{r}}$ and clock bias drift f . For this reason, an additional step was implemented to estimate velocity and clock bias drift, based on the rate of change of the receiver position and clock bias, respectively.

Algorithm 3.2 of Appendix A summarizes the steps used to solve for the GPS SPS at each time step for which / when measurement updates are available.

3.4 RDEKF Algorithm

The RDEKF algorithm tracks states for the satellite's position and velocity as well as the GPS receiver's clock. The satellite state vector \mathbf{x}_p is comprised of the 3D ECEF position and velocity vectors \mathbf{r} and $\dot{\mathbf{r}}$, respectively. The receiver clock state vector \mathbf{x}_c is composed of the clock bias b and clock drift f . The final 8×1 state vector is:

$$\mathbf{x} = [\mathbf{x}_p \quad \mathbf{x}_c]^T = [\mathbf{r} \quad \dot{\mathbf{r}} \quad b \quad f]^T.$$

Similar to the Kalman filter, the EKF is composed of a prediction phase and an update phase. In the prediction phase, the *a priori* state estimate $\hat{\mathbf{x}}_k^-$ and estimate covariance \mathbf{p}_k^- are generated for the current timestep t_k using initial conditions from the previous time step t_{k-1} (the prior). The prediction phase uses a linear clock model and Cowell's method to propagate the clock states and satellite states forward in time, respectively.

Two force models are investigated in this research: 1. a two-body orbit model (TB) and 2. a perturbed orbit model which includes consideration for the J2 Earth oblateness (TBJ2). Both models are investigated to evaluate the accuracy benefits for including the J2 Earth oblateness perturbation at the expense of increased computational complexity.

As previously discussed, the relative equations of motion for a satellite in orbit are only valid in an inertial reference frame. For this reason, the satellite states are converted to ECI coordinates using MATLAB's `ecf2eci()` function at the beginning of the prediction phase [76]. Afterwards, the states are converted back to ECEF coordinates using MATLAB's `eci2ecf()` function [77].

In the update phase, the *a posteriori* state estimate $\hat{\mathbf{x}}_k^+$ and estimate covariance \mathbf{p}_k^+ are generated based on a linear blend of the *a priori* state and measurements from the GPS sensor. Similar to the GPS SPS algorithm discussed in Section 3.3.2, the RDEKF algorithm uses a pseudorange-based GPS observation model.

Detailed descriptions of the RDEKF prediction phase and update phase are given in Section 3.4.1 and Section 3.4.2, respectively. Section 3.4.3 describes how the process and measurement noise covariance matrices are used to tune the RDEKF algorithm. A final summary of the RDEKF algorithm is given in Section 3.4.4 which provides some insight into its practical implementation into the ODS simulator developed in this thesis.

3.4.1 Prediction Phase

In the prediction phase, the state estimate and estimate covariance from the previous time step t_{k-1} are propagated forward by sample time Δt to the current time step t_k . For the clock states \mathbf{x}_c , the *a priori* state estimate is described in Equation 3.7 [7]:

$$\bar{\hat{\mathbf{x}}}_{c_k} = \boldsymbol{\phi}_c \bar{\hat{\mathbf{x}}}_{c_{k-1}} + \mathbf{w}_c \quad (3.7)$$

where,

$$\mathbf{x}_c = \begin{bmatrix} b \\ f \end{bmatrix} \quad \boldsymbol{\phi}_c = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}.$$

In Equation 3.7, $\boldsymbol{\phi}_c$ is the state transition matrix described in [7] and \mathbf{w}_c is the unmodelled process noise for the receiver clock.

For the satellite states \mathbf{x}_p , the *a priori* state estimate is calculated from Cowell's method. For this step, the derivative of the satellite state vector $\dot{\mathbf{x}}_p$ for the previous state estimate is generated using the relative equation of motion outlined in Section 2.1.3.

The a priori estimate covariance may then be calculated according to Equation 3.8:

$$\bar{\mathbf{p}}_k = \boldsymbol{\phi} \bar{\mathbf{p}}_{k-1} \boldsymbol{\phi}^T + \mathbf{Q} \quad (3.8)$$

where,

$$\boldsymbol{\phi} = \begin{bmatrix} \boldsymbol{\phi}_p & 0 \\ 0 & \boldsymbol{\phi}_c \end{bmatrix}.$$

In Equation 3.8, $\boldsymbol{\phi}_p$ represents the satellite state transition matrix which is linearized about the previous state estimate and \mathbf{Q} represents the process noise covariance matrix. The satellite state transition matrix is calculated in Equation 3.10 as the discretized form of the Jacobian matrix \mathbf{F} from Equation 3.9 [78]:

$$\mathbf{F} = \frac{\partial \dot{\mathbf{x}}_p}{\partial \mathbf{x}_p} \quad (3.9)$$

$$\boldsymbol{\Phi}_p \approx \mathbf{I} + \mathbf{F}\Delta t + \frac{1}{2}\mathbf{F}^2\Delta t^2 \quad (3.10)$$

where \mathbf{I} is the identity matrix. Detailed derivations of the state transition matrix for the TB and TBJ2 force models are given in [78].

Algorithm 3.3 in Appendix A summarizes the steps outlined in this section for the RDEKF prediction phase.

3.4.2 Update Phase

The RDEKF update phase described in this subsection is based on the derivations provided in [7]. In the update phase, the *a posteriori* state estimate $\hat{\mathbf{x}}_k^+$ and estimate covariance \mathbf{p}_k^+ are generated based on a linear blend of the *a priori* state and measurement data from the GPS sensor.

Using the *a priori* state estimate, the expected pseudorange $\hat{\rho}_{s_i}$, measurement residual $\Delta\rho_{s_i}$, and estimated line-of-sight vector $\hat{\mathbf{l}}_i$ are calculated for each tracked GPS satellite as per Equations 3.11 – 3.13, respectively:

$$\hat{\rho}_{s_i} = |\mathbf{r}_{s_i} - \hat{\mathbf{r}}_k^-| + c\hat{b}_k^- \quad (3.11)$$

$$\Delta\rho_{s_i} = \rho_{s_i} - \hat{\rho}_{s_i} \quad (3.12)$$

$$\hat{\mathbf{l}}_i^T = \frac{\mathbf{r}_{s_i} - \hat{\mathbf{r}}_k^-}{|\mathbf{r}_{s_i} - \hat{\mathbf{r}}_k^-|}. \quad (3.13)$$

In Equations 3.11 – 3.13, ρ_{s_i} and \mathbf{r}_{s_i} represent the pseudorange and ECEF satellite position observables, respectively, generated according to the GPS sensor model described in Algorithm 3.1 of Appendix A.

For n tracked satellites, the $n \times 1$ measurement residual vector $\Delta\boldsymbol{\rho}$ and $n \times 8$ observation matrix \mathbf{H} can then be constructed as follows:

$$\Delta\boldsymbol{\rho} = \begin{bmatrix} \Delta\rho_{s_1} \\ \Delta\rho_{s_2} \\ \vdots \\ \Delta\rho_{s_n} \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} -\hat{\mathbf{l}}_1^T & 0 & 0 & 0 & 1 & 0 \\ -\hat{\mathbf{l}}_2^T & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\hat{\mathbf{l}}_n^T & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Finally, the *a posteriori* state estimate and estimate covariance are calculated according to Equations 3.14 – 3.16:

$$\mathbf{K} = \mathbf{p}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{p}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (3.14)$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}\Delta\boldsymbol{\rho} \quad (3.15)$$

$$\mathbf{p}_k^+ = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{K}\mathbf{H}(\mathbf{I} - \mathbf{K}\mathbf{H})^T + \mathbf{K}\mathbf{R}\mathbf{K}^T. \quad (3.16)$$

In Equation 3.14, \mathbf{K} is the Kalman gain term which assigns more weight to the dynamic or the GPS kinematic solution depending on the *a priori* estimate covariance and measurement noise covariance matrix \mathbf{R} .

Algorithm 3.4 in Appendix A summarizes the steps outlined in this section for the RDEKF update phase.

3.4.3 Filter Tuning

The process and measurement noise covariance matrices determines whether the RDEKF state estimate will tend towards a dynamic solution or a kinematic one. The process noise covariance matrix relates to the quality of the process model used in the prediction phase; this matrix is used to tune the RDEKF algorithm to achieve optimal performance in terms of estimation accuracy and continuity [7]. The measurement noise covariance matrix is typically generated based on estimated parameters which describe the quality of the GPS receiver observables.

The process noise covariance matrix is composed of both satellite state and clock state parameters as shown in the following [7]:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_p & 0 \\ 0 & \mathbf{Q}_c \end{bmatrix}$$

where,

$$\mathbf{Q}_p = \begin{bmatrix} \sigma_r^2 I_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \sigma_{\dot{r}}^2 I_{3x3} \end{bmatrix} \quad \mathbf{Q}_c = \begin{bmatrix} s_b \Delta t + S_f \frac{\Delta t^3}{3} & S_f \frac{\Delta t^2}{2} \\ S_f \frac{\Delta t^2}{2} & S_f \Delta t \end{bmatrix}.$$

The satellite state noise matrix \mathbf{Q}_p is a 6×6 diagonal matrix which is generated based on the position state variance σ_r^2 and velocity state variance $\sigma_{\dot{r}}^2$. The clock state standard deviation matrix \mathbf{Q}_c used in this research was generated based on the model described in [7]. In this model, s_b and S_f represent white noise spectral amplitudes for the GPS receiver's clock.

The measurement noise covariance matrix is equal to the following $n \times n$ diagonal matrix:

$$\mathbf{R} = \sigma_\rho^2 \mathbf{I}_{n \times n}$$

where σ_p^2 is a constant term used to represent the expected pseudorange measurement variance.

Filter parameters used in this research were found through a manual tuning campaign based on initial estimates from [7,79] which use similar RDEKF algorithm implementations to the one described here.

3.4.4 Algorithm Summary

The RDEKF algorithm requires initial estimates of its state \hat{x}_0 and estimate covariance p_0 to begin. The accuracy of the initial state estimate can significantly affect convergence as illustrated in later sections; however, without too much loss of generality, the estimate covariance can be initialized as the identity matrix.

Once valid initial conditions are provided, the algorithm generates regular state estimates at a user defined time step Δt . The state prediction phase is used for every time step; however, the state update phase may only be used for time steps where GPS measurements are available. Furthermore, for applications with limited onboard resources, use of the GPS sensor may be limited. In these cases, a user defined measurement time step defines the time between each measurement update.

In between measurement updates, the state estimate will be based on a purely dynamic solution (the process model) which only diverges with time. Therefore, the quality of the process model used in the state prediction phase will dictate the maximum allowable measurement time step for a given accuracy requirement.

A flowchart summarizing the overall RDEKF algorithm is provided in Figure 38 of Appendix B.

3.5 ODS Simulation Testing

In this subsection, the SGP4, GPS SPS, and RDEKF navigation algorithms are evaluated based on tests completed with the ODS simulator developed in this thesis. Each test is initialized by specifying the selected dataset timeframe, navigation algorithm, and any other configurable parameters which may be technique dependent. The simulation time was supplied to the navigation algorithms based on fixed 1-second time steps from a user-specified start time. For each simulation time step, an output array containing the satellite state information is stored for post-processing. Furthermore, time steps when GPS

measurements were available are tracked and recorded. A flowchart describing the overall ODS simulator process is shown in Figure 39 of Appendix B.

Algorithms were evaluated based on their overall position and velocity estimation accuracy and estimated GPS duty cycle. 3D root-mean-square-error (RMSE) was used as the primary performance metric for evaluating estimation accuracy; however, the maximum 3D position and velocity errors for each simulation test were also considered as secondary metrics. As previously discussed (Section 2.3.2), a 3-minute TTFF was assumed for GPS warm-starts based on reported experimental results [16,26]. For measurement timesteps of 180 seconds or less, a 100% GPS duty cycle was assumed; otherwise, the duty cycle was calculated as in Equation 3.17:

$$GPS \text{ duty cycle} = 100\% \times \frac{180 \text{ s}}{\Delta t_m} \quad (3.17)$$

where Δt_m is the prescribed measurement timestep in seconds.

3.5.1 SGP4 Technique Evaluation

The testing completed in this section illustrates the instantaneous and long-term estimation accuracy of the SGP4 algorithm with a single TLE dataset. To start, the SGP4 algorithm estimates the satellite's state with a single TLE over the 24-hour period from Day 226 of 2018.

The 3D position and velocity RMSE for this test are shown in Figure 1. As expected, the SGP4 algorithm provides fairly coarse estimation accuracy with a final RMSE of 821.4 m and 3.95 m/s for 3D position and velocity, respectively.

To illustrate the benefit of mean orbital elements, a second test was conducted where the SGP4 algorithm estimated the satellite's state with a single TLE over the 14-day period from Day 226 to 239 of 2018. For comparison, another test was conducted over the same period with daily TLE updates.

The resulting 3D position and velocity RMSE for both tests are shown in Figure 2. The SGP4 estimation accuracy is approximately constant over the 14-day period when provided with daily TLE updates; however, as expected, performance slowly degrades as the time since TLE epoch increases.

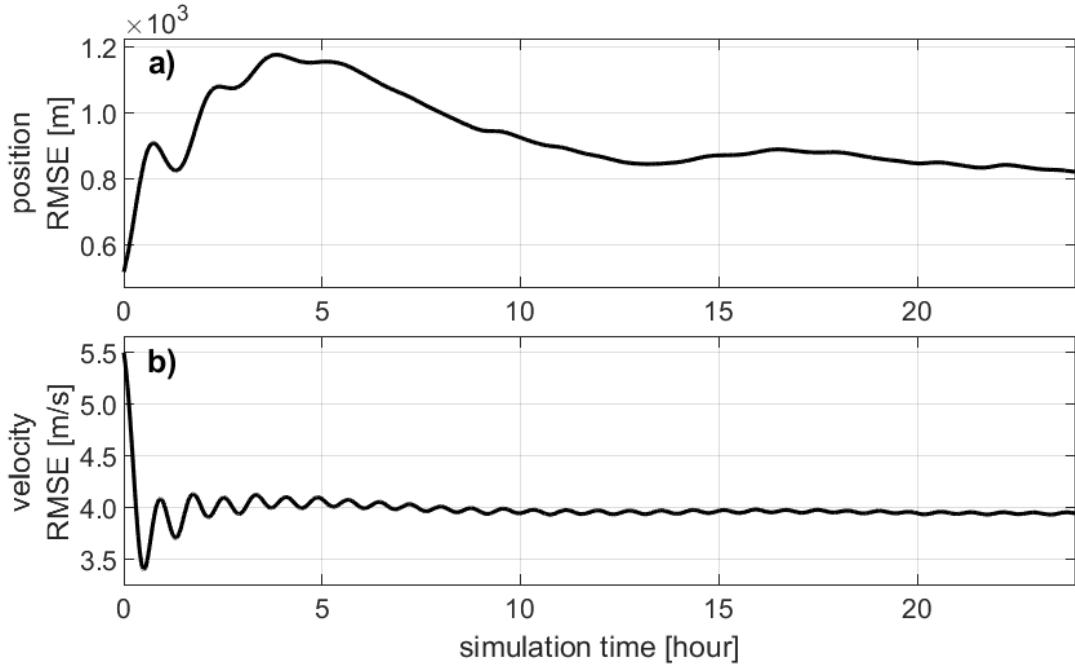


Figure 1. SGP4 24-hour 3D RMSE: a) position and b) velocity RMSE vs. time since simulation start for Day 226 of 2018. Accuracy is coarse, as expected.

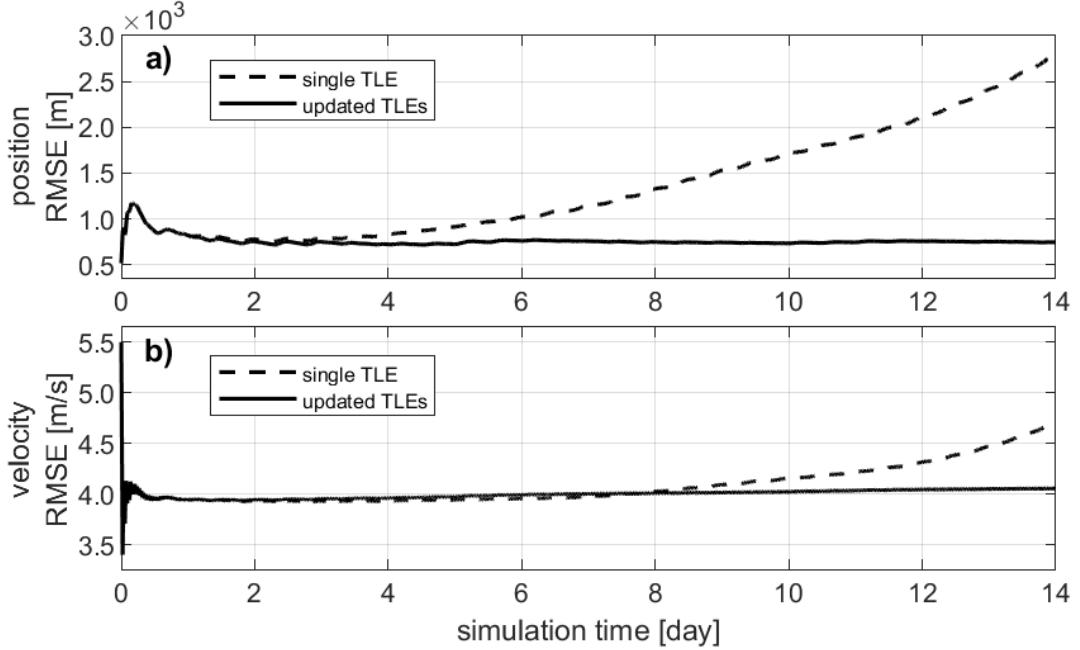


Figure 2. SGP4 14-day 3D RMSE: a) position and b) velocity RMSE vs. time since simulation start for Day 226 to 239 of 2018. The SGP4 algorithm is fairly resilient to infrequent TLE updates making it a good backup strategy for onboard orbit determination.

With only a single TLE over a 14-day period, the SGP4 algorithm produced a final RMSE of 2.802 km and 4.71 m/s for 3D position and velocity, respectively. This accuracy

may be insufficient for regular onboard GNC operations; however, the SGP4 algorithm does not require GPS and is resilient to infrequent TLE updates. For these reasons, the SGP4 algorithm should be used when only coarse accuracy (< 3 km) is required or the GPS is unavailable.

In the following subsection, similar tests are performed with the custom GPS SPS algorithm implementation in the ODS simulator.

3.5.2 GPS SPS Technique Evaluation

The GPS SPS was calculated over the 24-hour period from Day 226 of 2018. Raw GRACE-FO GPS pseudorange measurements were only available at a rate of 0.1 Hz which limited the availability of SPS state estimates to once every 10 seconds.

The GPS SPS generated several large outliers which are likely attributed to instances with poor GPS satellite observation geometry. For onboard ODS applications, outliers should be identified and removed in real-time. In this research, outliers were removed in post-processing. MATLAB's `isoutlier()` function removed data points which resulted in 3D position or velocity errors above the 99.95 percentile of the total error distribution [80].

In Figure 3, the resulting 3D position and velocity errors for the 24-hour GPS SPS test are shown for both the raw and post-processed (outlier-removed) solutions. There were four outliers detected over the entire 24-hour testing period. The remaining errors resulted from the limited observation model used in the GPS SPS algorithm and high-frequency receiver noise.

The final RMSE for the post-processed 3D position and velocity errors are 21.54 m and 42.32 m/s, respectively. The position accuracy of the GPS SPS is considerably better than the results from the SGP4 algorithm testing in Section 3.5.1. On the other hand, the velocity accuracy is worse; this may be attributed to the limited observation model used for the GPS SPS in this research which approximates velocity with Euler's method.

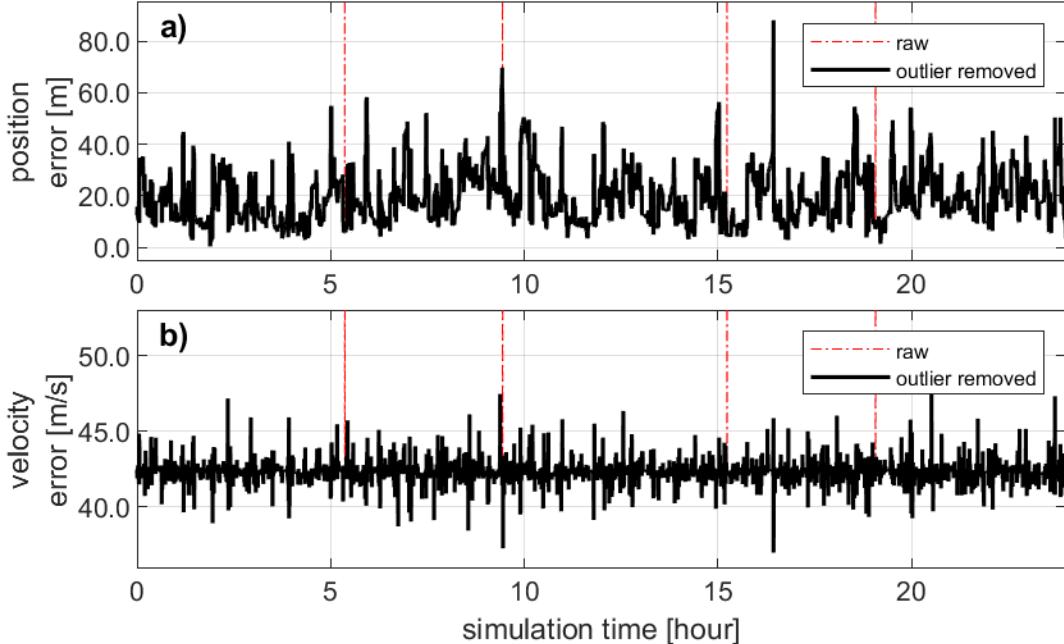


Figure 3. GPS SPS 24-hour 3D error: a) position and b) velocity error vs. time since simulation start for Day 226 of 2018. Position accuracy is improved considerably compared to the results from the SGP4 algorithm testing in Section 3.5.1; however, velocity accuracy is worse due to limitations of the GPS SPS observation model implemented in this research.

The GPS SPS is not recommended for real-time onboard orbit determination due to its inability to estimate the satellite's state in-between GPS measurements; however, it can produce accurate position state estimates (< 200 m) for geocoding payload data or supply initial conditions to the RDEKF algorithm. Furthermore, alternative GPS observation models can be investigated if more accurate position or velocity state estimates are required.

In the following subsection, different RDEKF algorithm configurations are tested with the ODS simulator.

3.5.3 RDEKF Technique Evaluation

The performance of the RDEKF algorithm depends on its setup configuration. Configurations of the RDEKF algorithm are primarily based on the selected process model, observation model, and GPS duty cycle (as in EKFs in general). Additionally, to begin estimating the satellite state, the RDEKF algorithm requires a set of initial conditions (IC). The purpose of the experiments completed in this section are two-fold: 1. to test how ICs from different sources affect the convergence speed and 2. to evaluate the performance of

RDEKF algorithm given the selected process model and GPS duty cycle. All tests completed in this section use GRACE-FO experimental measurements from Day 226 of 2018.

Position and velocity ICs are generated from the previously described navigation algorithms. Ideally the GPS navigation solution should be used to provide ICs to the RDEKF algorithm due to its superior accuracy compared to the SGP4 algorithm; however, as shown in Section 3.5.2, the velocity estimation accuracy of the GPS SPS implementation used in this research is relatively poor. For this reason, the SGP4 algorithm was also tested.

The three IC configurations considered are based on the SGP4 algorithm alone, the GPS SPS algorithm alone, and a combination of position ICs and velocity ICs from the GPS SPS and SGP4 algorithms, respectively. The resulting 3D position and velocity errors for the RDEKF IC tests are shown in Figure 4.

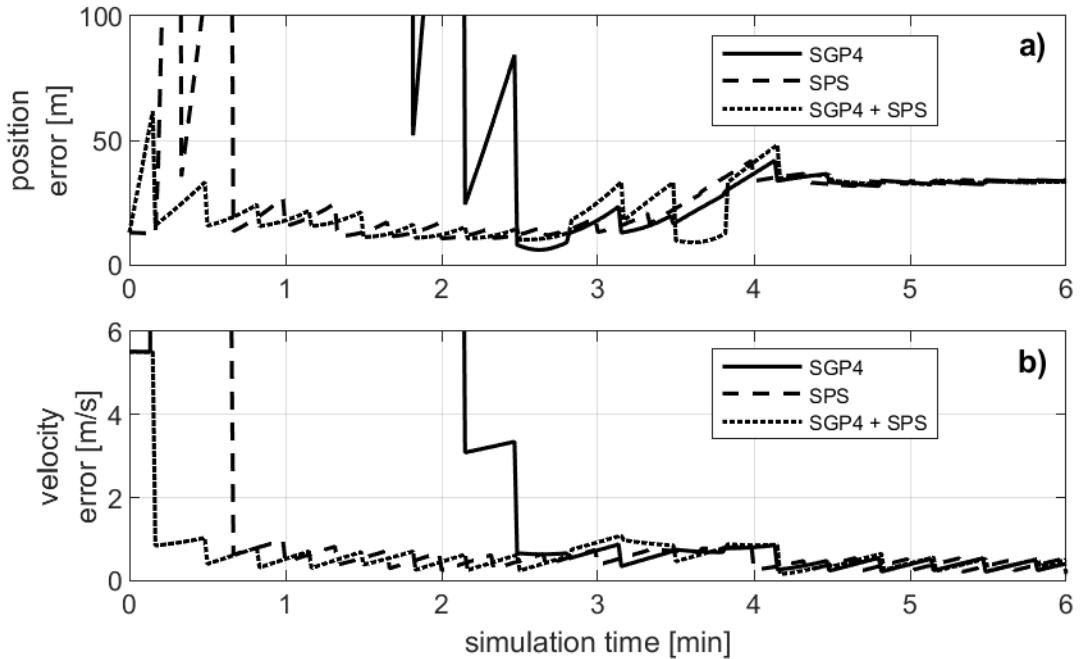


Figure 4. RDEKF initial condition (IC) experiment results: a) position and b) velocity error since simulation start for Day 226 of 2018. The fastest convergence was observed when position ICs were supplied by the GPS SPS algorithm and velocity ICs by the SGP4 algorithm; this is compared to the first two test cases which were provided initial conditions from either SGP4 or GPS SPS.

All three RDEKF configurations eventually converge to approximately the same values; however, there is a noticeable difference in the time (rate) it takes each configuration to converge. Supplying ICs from the SGP4 algorithm alone resulted in a maximum position error of over 10 km and the slowest convergence time at $t = 2.15$

minutes. Supplying ICs from the GPS SPS algorithm alone resulted in a maximum position error of 374.5 m and a convergence time at $t = 0.67$ minutes. The third configuration used the best estimates from both the GPS SPS and the RDEKF algorithms and yielded the best results, as expected, with a maximum position error of 61.7 m and the fastest convergence time at $t = 0.17$ minutes; this IC configuration was selected for the remainder of this thesis.

Moving forward, a sensitivity analysis was conducted to evaluate the RDEKF algorithm's performance as a function of the selected process model and measurement time step in its setup configuration. As discussed in Section 3.4, the two process models tested in this research include the two-body (TB) and two-body J2 (TBJ2) models. RDEKF algorithm configurations were tested over a 3-hour period with measurement time steps between 10 s and 1800 s (30 minutes). Furthermore, the SGP4 algorithm was tested over the same 3-hour period as a baseline for comparison; the final 3D position and velocity RMSE for this test were 1088 m and 3.96 m/s, respectively.

The detailed results from this experiment, including the GPS duty cycle and accuracy performance metrics (i.e. RMSE and maximum error), are provided in Tables 13 and 14 of Appendix C.1. The TB process model RDEKF configuration was only tested with measurement timesteps up to 450 s (7.5 minutes) at which point its final 3D position RMSE exceeded that of the SGP4 baseline test.

The final 3D position and velocity RMSE for each RDEKF simulation test configuration are plotted against the selected measurement timestep in Figure 5 along with the SGP4 baseline test results. Both RDEKF process models perform similarly for small measurement time steps; however, the accuracy of the TB process model quickly degrades with increasing measurement time steps. The TBJ2 process model is much more resilient to infrequent GPS measurements and maintains superior estimation accuracy to the SGP4 algorithm even when 30-minute measurement time steps are used.

The best performance was observed for the TBJ2 RDEKF algorithm configuration with a 100% GPS duty cycle which produced a final 3D position and velocity RMSE of 19.8 m and 0.29 m/s, respectively. Similar performance may be achieved with the TB process model to reduce onboard processing; however, this solution is less tolerant to GPS outages. If onboard power is limited, the TBJ2 process model should be implemented with a reduced GPS duty cycle.

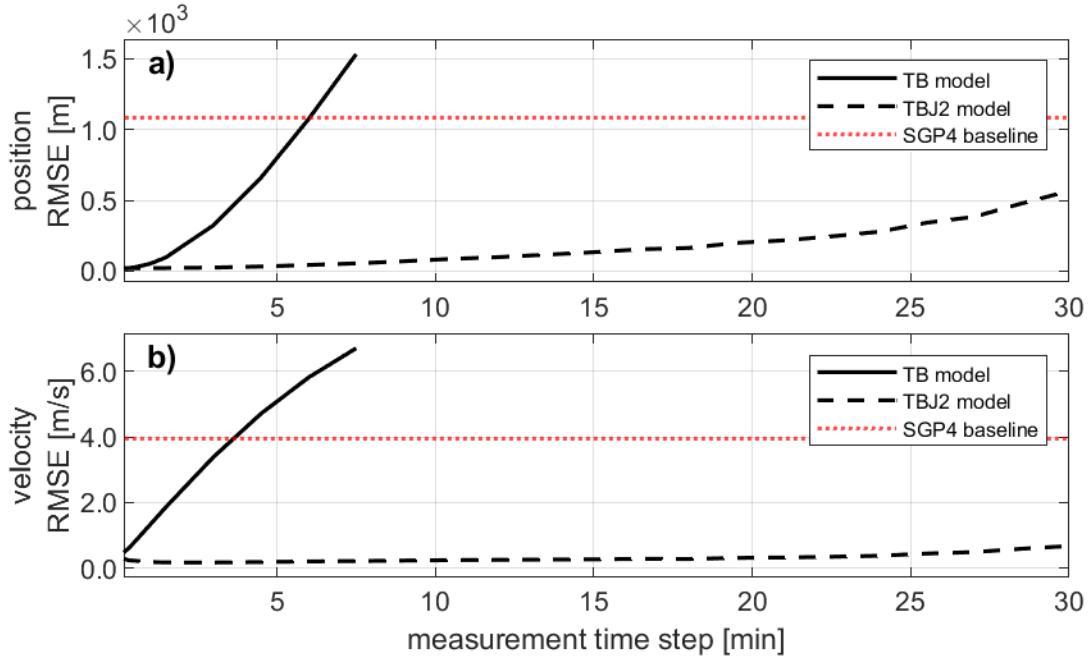


Figure 5. RDEKF model configuration experiment results: a) position and b) velocity 3-hour RMSE vs. GPS measurement time step for Day 226 of 2018. RDEKF configurations with the two-body J2 (TBJ2) process model are more resilient to infrequent GPS measurements; however, the two-body (TB) process model configurations are more efficient and produce comparable accuracy when GPS is consistently available.

It is important to note that the results shown here reflect the conditions from a relatively small dataset. To validate any single RDEKF configuration, it should be tested against multiple datasets which are indicative of the expected mean and worst-case operating conditions.

In the following subsection, the SGP4, GPS SPS, and SGP4 navigation algorithms are compared, and recommendations made for CubeSat Earth observation missions with limited onboard resources.

3.5.4 ODS Techniques Comparison

In this subsection, the ODS techniques are compared directly based on their performance for tests performed in the ODS simulator for the same 24-hour period from Day 226 of 2018. In addition to the SGP4 and SPS navigation algorithms, three configurations of the RDEKF algorithm were tested: a TB model with 10 s measurement timesteps (TB-10s), a TBJ2 model with 10 s measurement timesteps (TBJ2-10s), and a TBJ2 model with 810 s measurement timesteps (TBJ2-810s).

The results of these tests are summarized in Table 1.

Table 1. ODS techniques comparison summary. The SGP4 algorithm produces coarse estimation accuracy and does not require GPS. The TB-10s and TBJ2-10s RDEKF algorithm configurations produce comparable accuracy to the GPS SPS and are more robust to GPS outages. The TBJ2-810s RDEKF algorithm configuration compromises between estimation accuracy and GPS power consumption.

solver	GPS duty cycle [%]	3D position error [m]		3D velocity error [m/s]	
		RMSE	max error	RMSE	max error
SGP4	-	821.4	1562	3.95	5.82
SPS	100.0	21.54	88.15	42.3	48.3
TB-10s	100.0	22.88	364.9	0.558	19.0
TBJ2-10s	100.0	22.38	364.0	0.390	19.0
TBJ2-810s	22.0	73.72	580.1	0.155	0.814

As previously discussed in Section 2.3.5, positional accuracies less than 200 m are considered for fine orbit determination and those greater than 200 m but less than 3 km are considered for coarse orbit determination, where the final 3D RMSE is used to represent the overall estimation accuracy.

As expected, the SGP4 algorithm achieved fairly coarse estimation accuracy with a final 3D RMSE of 821.4 m and 3.95 m/s for position and velocity, respectively. However, this algorithm does not require GPS and is also robust to infrequent TLE updates making it one of the safest options for coarse orbit determination.

The GPS SPS algorithm produced a final 3D RMSE of 21.54 m and 42.3 m/s for position and velocity, respectively. The position estimation accuracy of the GPS SPS is considerably better than the SGP4 algorithm; a more robust observation model implementation will improve accuracy further (particularly for velocity which was estimated using Euler's method in this research) [35]. The GPS SPS is fairly simple to implement compared to the RDEKF algorithm; however, state estimates are dependent on the availability of GPS measurements.

The RDEKF algorithm is a good alternative to purely kinematic solutions like the GPS SPS. Both the TB and TBJ2 process models produce comparable position accuracy to the GPS SPS with 100% GPS duty cycle. Furthermore, the TBJ2 model is less sensitive to infrequent GPS measurements. For example, the TBJ2 model RDEKF configuration with a 22% GPS duty cycle resulted in a final 3D RMSE of 73.72 m and 0.155 m/s for

position and velocity, respectively. However, the RDEKF algorithm uses numerical integration at every time step which means increasing force model complexity results in significant onboard processing.

Both the GPS SPS and RDEKF navigation algorithms use GPS measurements and are therefore sensitive to instances with poor GPS satellite viewing geometry. During testing, the GPS SPS resulted in several large outliers as a result of these instances; similar outliers were observed for the RDEKF algorithm 24-hour simulation tests. These instances were infrequent and were therefore removed during post-processing to calculate the final 3D RMSE for tests completed in the ODS simulator. For onboard ODS applications, outliers should be identified and removed in real-time.

Each of the tested navigation algorithms has advantages and disadvantages for onboard orbit determination. For coarse orbit determination, or as a backup when GPS is unavailable, the SGP4 algorithm should be used. If onboard POD is not required for navigational purposes, the GPS SPS can provide fine positional estimation accuracy for geocoding payload data. The RDEKF algorithm should be used for continuous onboard POD. If GPS is consistently available, the TB process model configuration reduces onboarding processing; however, the TBJ2 process model configuration is more tolerant to GPS outages and/or duty cycling. Finally, the RDEKF algorithm should be supplied position and velocity ICs from the GPS SPS and RDEKF algorithms, respectively, for the best performance.

In the following chapter, a technique to automatically switch between ODS techniques during Earth-target flybys is developed to minimize use of onboard resources thus making it suitable for a CubeSat implementation.

Chapter 4. OGNS Development

The testing completed in Chapter 3 illustrates the advantages and disadvantages of the SGP4, GPS SPS, and RDEKF navigation algorithms. Ideally, an onboard ODS will use some form of reduced dynamic technique with continuous GPS measurement updates. However, this may be impractical for CubeSat platforms with limited onboard resources. Furthermore, constant POD may be unnecessary for some applications.

For instance, the LORIS mission objective is to take pictures of the Nova Scotia coastline; however, the satellite will only be within range of the target during a small fraction of its time spent in orbit. To conserve onboard power, it is more efficient to only use the GPS as needed by the onboard ADCS during target flybys. For the remainder of the time spent in orbit, the GPS could be turned off and the SGP4 algorithm used for coarse orbit determination.

The difficulty of this approach is knowing when to switch between coarse and fine ODS strategies. Instantaneous switching is not practical due to the expected hot-start delays which occur whenever the GPS receiver is turned back on. Furthermore, the ADCS system may need sufficient time to ensure the camera is pointed towards Earth during the target flyby. It is therefore necessary to predict target flybys with sufficient time in advance to schedule onboard GNC operations.

In this chapter, a combined orbital guidance and navigation system (OGNS) is developed which uses an onboard target flyby prediction algorithm to automatically switch between coarse and fine ODS strategies. As before (Section 3.5.4), positional accuracies less than 200 m are considered fine and those greater than 200 m but less than 3 km are considered coarse, where the final 3D RMSE is used to represent the overall estimation accuracy.

4.1 Target Flyby Prediction

In this research, targets are represented as fixed geodetic coordinates on the Earth's surface. A flyby can therefore be defined as the period during which the satellite is visible from the target's ground location and vice versa. The satellite is considered visible when it crosses the target's horizon as shown in Figure 6.

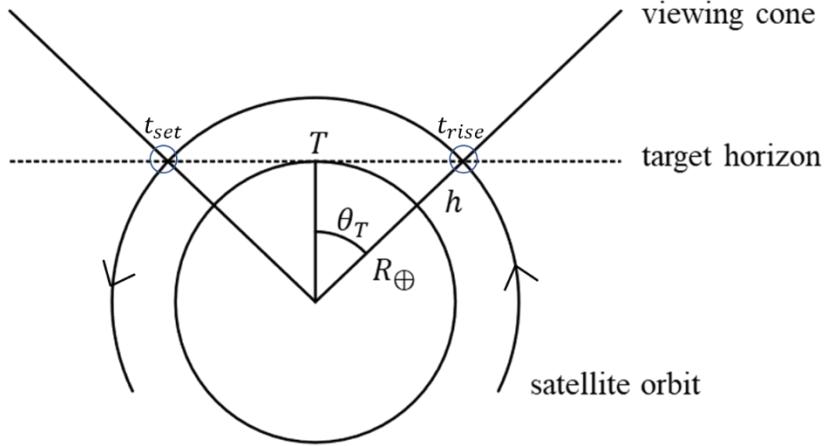


Figure 6. 2D Projection of satellite in Earth orbit illustrating target-satellite visibility criterion.

In Figure 6, θ_T represents the target visibility angle which subtends the satellite's nominal orbit radius $R_{\oplus} + h$ at the target's horizon. The target visibility angle can be calculated using Equation 4.1:

$$\theta_T = \cos^{-1} \left(\frac{R_{\oplus}}{R_{\oplus} + h} \right). \quad (4.1)$$

The target visibility angle can be used with the target's latitude λ_T and longitude ϕ_T to generate a 3D viewing cone which represents the target flyby region in geodetic coordinates.

The visibility problem essentially boils down to solving for the rise time t_{rise} and set time t_{set} which define the first and last instant that the satellite is above the target, respectively. In Section 4.1.1, a literature review of techniques to predict the rise time and set time is given. Afterwards, a two-step target flyby prediction algorithm is presented. In the first step, a coarse target prediction algorithm predicts the integral number of remaining orbits until a target flyby occurs. Then, a fine target prediction algorithm estimates the rise time and set time for the identified target flyby. The coarse and fine target prediction algorithms are discussed in Section 4.1.2 and 4.1.3, respectively.

4.1.1 Satellite Visibility Prediction Literature Review

Possibly the simplest approach to solve the satellite visibility problem would be the brute force method. In this approach, the satellite’s state is propagated forward in time until the next target flyby can be identified. The brute force method generates accurate predictions with a good orbit model; however, it is not well suited for onboard applications with limited processing power.

One of the first successful applications of a closed-form solution to the satellite visibility problem [81]. The author presents a single transcendental equation to solve for rise and set time as a function of the satellite’s eccentric anomaly. Newton’s method is applied once per orbit to solve for the rise and set eccentric anomalies with the assumption of an oblate Earth.

In [82], a similar approach is presented which includes consideration for additional long-term periodic and secular variations of the satellite’s orbital elements. Their method begins with a coarse search phase which predicts the remaining number of orbits before a flyby will occur at the target latitude line. Then, the refined transcendental equation solves for the final rise and set times. Their method produces accuracies on the order of a few seconds for predictions made over one month ahead.

Alternative “fast orbit prediction” techniques have been studied [83,84,85]. These approaches are similar in that they present computationally efficient techniques for accurately predicting upcoming target flybys. The implementation of accurate target flyby prediction algorithms with large look-ahead periods is an important development for improving satellite onboard autonomy. To switch between onboard ODS techniques, a modest accuracy requirement of several seconds can be sufficient if a time buffer is added to the beginning and end of the predicted target flyby rise and set times, respectively.

4.1.2 Coarse Target Prediction

The coarse target prediction algorithm is based on the coarse search phase from [82]. This algorithm estimates the integral number of orbits remaining until the satellite is within range of its target at the target latitude line (TLL).

For each full orbit, the satellite's longitude will be displaced to the west by approximately the same amount which is equal to the satellites orbital period P_s multiplied by Earth's rotation rate ω_{\oplus} . This pattern is illustrated in Figure 7.

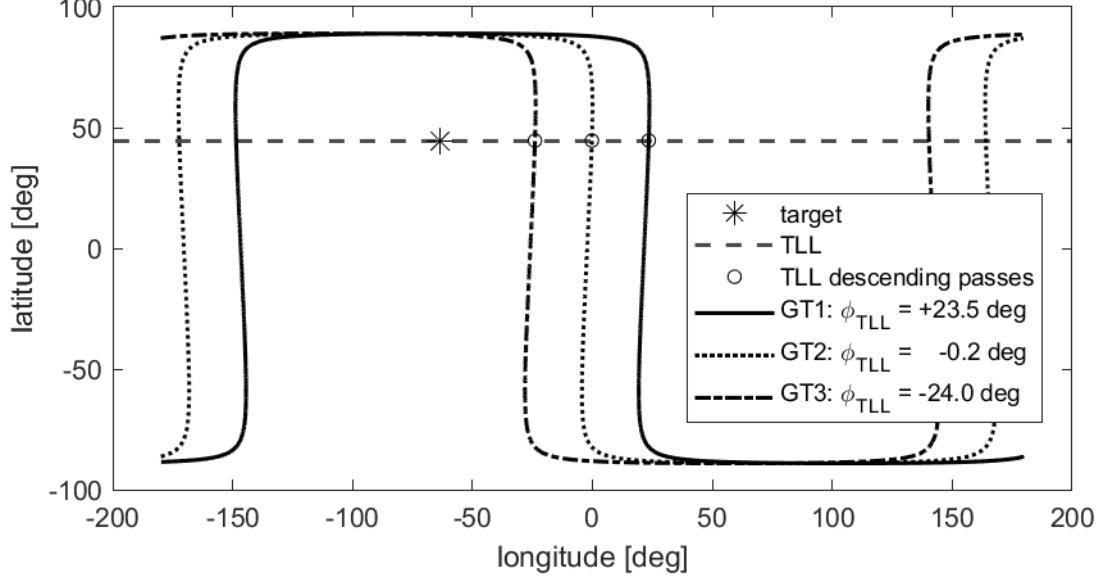


Figure 7. GRACE-FO satellite ground tracks illustrating descending passes of the target latitude line for Halifax, Canada. The longitudinal displacement between ground tracks is approximately constant.

In Figure 7, 2D ground track projections illustrate the continuous trajectory of the GRACE-FO satellite plotted together with a target representing the geodetic coordinates of Halifax, Nova Scotia. The locations of each subsequent descending pass of the TLL are indicated.

At each TLL crossing event, the number of orbits remaining until a target flyby N is calculated according to Equation 4.2:

$$N = \left\lceil \frac{\Delta\phi_v}{\hat{P}_s \omega_{\oplus}} \right\rceil + 1 . \quad (4.2)$$

In Equation 4.2, $\Delta\phi_v$ is the longitudinal separation between the target viewing cone and the satellite and \hat{P}_s is the satellites estimated orbital period which is updated every orbit based on the time between the subsequent TLL crossing events.

A visual representation of the longitudinal separation for the GT1 TLL descending pass from Figure 7 is shown in Figure 8.

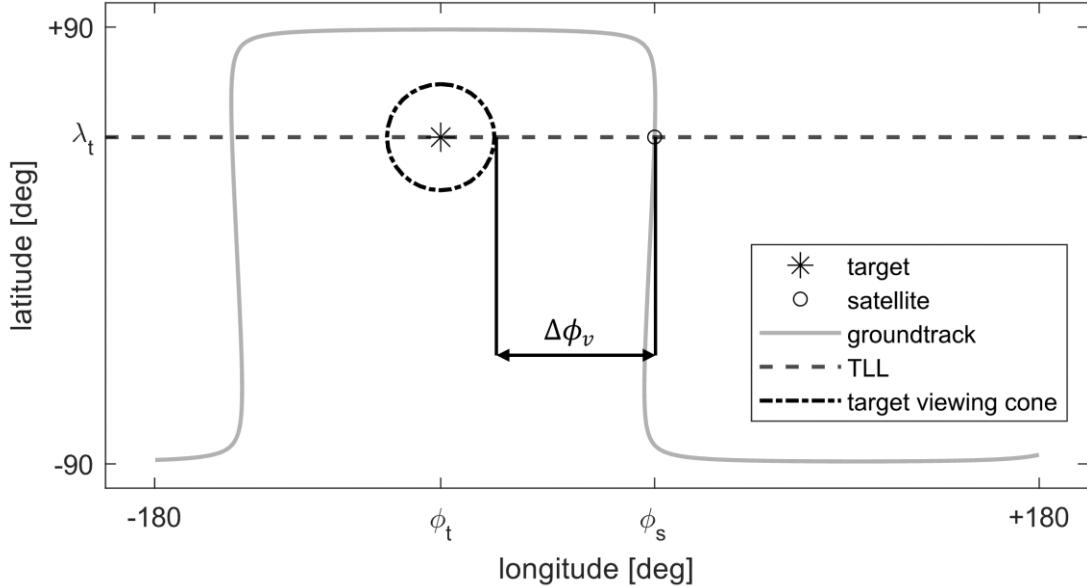


Figure 8. GRACE-FO satellite ground track illustrating the longitudinal separation from the target viewing cone during a descending pass of the target latitude line.

Algorithm 4.1 in Appendix A summarizes the coarse target prediction algorithm from [82] including how to calculate the longitudinal separation at TLL when the satellite is located east or west of the target viewing cone. In this research, a modified coarse target prediction algorithm was used which includes consideration for TLL crossings that occur within the target viewing cone. This step was necessary to account for repeat target flybys (i.e. target flybys which happen twice in a row). The modified coarse target prediction algorithm can be found in Algorithm 4.2 of Appendix A.

While the examples provided in this section only refer to descending passes of the TLL, the actual algorithm implementation is used to separately track and predict flyby events for both ascending and descending TLL passes.

4.1.3 Fine Target Prediction

The fine target prediction algorithm predicts the rise time and set time for a target flyby which is N orbits away; however, for simplicity, the fine target prediction algorithm is only used in this research when N is equal to one (i.e. a target flyby has been predicted for the upcoming TLL crossing). The coarse target prediction algorithm operates under the assumption of approximately constant ground track displacements for each orbit; a similar

assumption is used here to predict future satellite coordinates from the current state estimate.

The satellite's future longitude ϕ_s' can be predicted from its current longitude ϕ_s minus the predicted ground track displacement $\hat{P}_s \omega_{\oplus}$. Furthermore, the satellite's future latitude λ_s' will be approximately equal to its current latitude λ_s . The future coordinates of the satellite can then be used to determine whether the satellite will be within the target's viewing cone at the future time step t' . These steps are repeated for each time step that the satellite is within the latitude range $[\lambda_T - \theta_T, \lambda_T + \theta_T]$ to identify the rise time and set time for future target flybys. Algorithm 4.3 in Appendix A summarizes the fine target prediction algorithm described here.

4.2 OGNS Algorithm Implementation

The OGNS algorithm acts as a switch between coarse and fine ODS strategies based on target flyby predictions. At the beginning of each time step, the ODS switch decision block determines which ODS navigation algorithm to use. The logic for this decision block is based on feedback from the fine target prediction algorithm. If the current time step falls within the predicted flyby period, the fine ODS algorithm is used; otherwise, the coarse ODS algorithm is used. Then, the updated state estimate is converted to geodetic coordinates and the OGNS algorithm checks to see whether coarse or fine target predictions should be used for the current time step.

The decision block for the coarse target prediction algorithm determines if a TLL crossing has just occurred; if this is the case, the algorithm predicts the remaining number of orbits until the satellite will be within range of its target. The decision block for the fine target prediction algorithm then determines if a flyby will occur after the next full orbit. Unlike the coarse target prediction algorithm, the fine target prediction algorithm does not produce an instantaneous prediction; rather, it is applied at each time step to identify the rise and set times for the upcoming target flyby.

A flowchart illustrating the OGNS algorithm is shown in Figure 9. For the tests completed in this research, the SGP4 algorithm and RDEKF algorithm were used for coarse and fine orbit determination, respectively. During target flybys, the TB RDEKF algorithm configuration provided state estimates once per second and GPS measurements were

supplied once every 10 seconds. Outside of target flybys, the SGP4 algorithm provided state estimates once per second and the GPS was not used. Furthermore, a 60-second buffer was added to the beginning and end of each fine target flyby prediction period to allow the RDEKF algorithm time to converge with sufficient time in advance to the flyby.

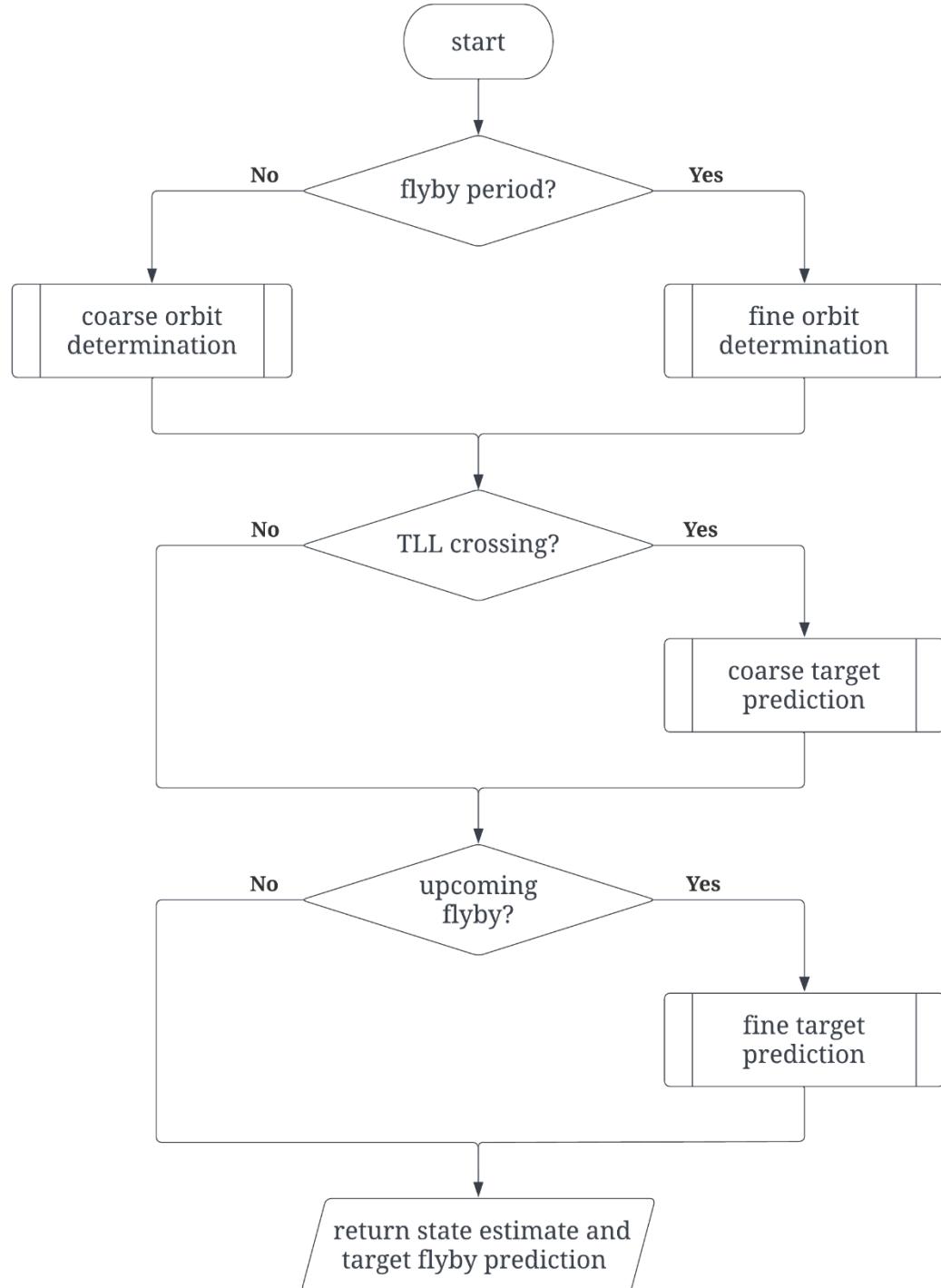


Figure 9. Orbital guidance and navigation system process used in this research.

4.3 OGNS Simulation Testing

The OGNS algorithm was tested for three different target locations using GRACE-FO data for the 4-day period from Day 226 to 229 of 2018. The same performance metrics from Chapter 3 were used here; however, only data from target flyby regions was selected to calculate the final state estimation accuracy considered in this chapter. Furthermore, GPS duty cycle was calculated based on the total GPS ‘on time’ (i.e. while the RDEKF is active, including the 3-minute TTFF assumption) as a percentage of the simulation duration.

The first test was for a target located in Halifax, Canada. During this test, there were 14 target flybys which were all successfully identified by the OGNS algorithm. In Figure 10, the 3D position state estimation error is shown for the first target flyby which lasted approximately 8 minutes.

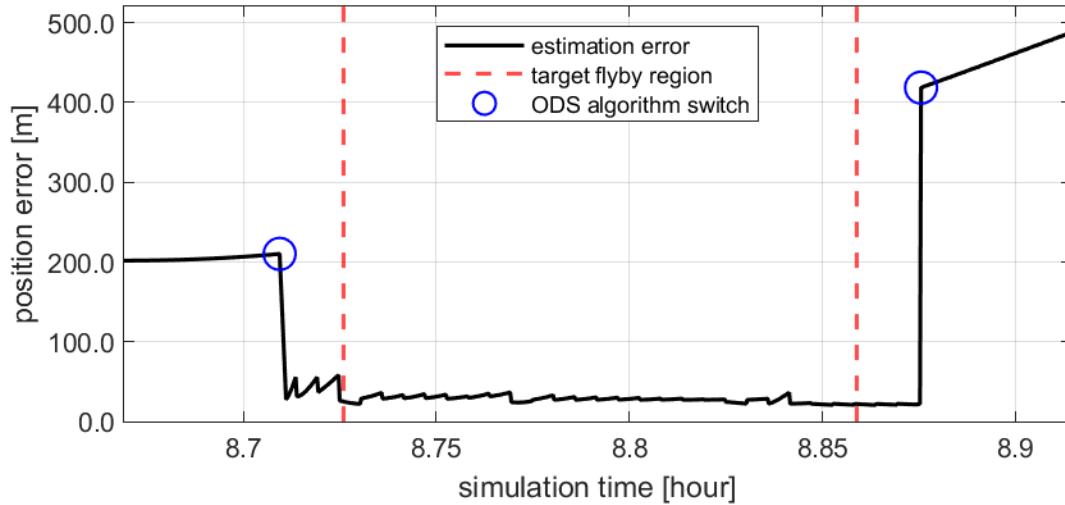


Figure 10. OGNS 3D position error vs. time for the first Halifax target flyby on Day 226 of 2018. The OGNS algorithm automatically switches from the SGP4 to the RDEKF navigation algorithm resulting in improved position estimation accuracy during the target flyby which lasted approximately 8 minutes.

In Figure 10, the switch from the SGP4 to the RDEKF navigation algorithm is clearly identified by the sharp decrease in error at $t = 8.71$ hours. The 3D RMSE position and velocity errors for the flyby region were 28.3 m and 0.43 m/s, respectively. Similar 3D position error plots for Halifax target flybys which occurred in the first 24 hours of the simulation are shown in Figure 40 of Appendix C.2.1. Furthermore, the summarized results of all 14 Halifax target flybys from the 4-day testing period can be found in Table 15 of Appendix C.2.2.

As previously discussed (Section 2.3.3), the performance of the RDEKF algorithm depends on the selected force model, where unmodelled perturbations reduce estimation accuracy. The most significant perturbations in LEO result from the non-spherical Earth and induce cyclic errors which correspond to the satellites ECEF position. To further validate the OGNS algorithm, two additional target locations were selected to ensure fine orbital accuracy (< 200 m) is achieved for flybys which occur in significantly different regions. Halifax is located in the northern and western hemispheres; Rio de Janeiro, Brazil and Mumbai, India were therefore selected to test target locations in the southern and eastern hemispheres, respectively. 3D position error plots for flybys which occurred in the first 24 hours of testing are shown in Figures 41 and 42 of Appendix C.2.1 for the Rio and Mumbai target tests, respectively. Summarized results from the tests can also be found in Tables 16 and 17 of Appendix C.2.2.

The OGNS algorithm successfully predicted 13 of the 14 flybys for the Rio target test. Outliers which occurred from instances with poor GPS satellite observation geometry were removed from 3 Rio target flybys as per the procedure outlined in Section 3.5.2 using MATLAB's *isoutlier()* function [80]. For the Mumbai target test, the OGNS algorithm successfully predicted 13 of the 14 flybys and no large outliers were detected. Furthermore, the missed flyby occurred within the first full orbit which resulted in the OGNS algorithm having insufficient information for target prediction.

The results for all three OGNS target tests are summarized in Tables 2 and 3.

Table 2. 96-hour OGNS testing results summary: part 1. The OGNS algorithm successfully predicted 40 of the 42 target flybys, automatically switched between SGP4 and RDEKF navigation algorithms, and minimized GPS duty cycle to less than 1% for the entire testing period.

target	flyby summary		GPS usage		
	total	missed	on time [min]	off time [min]	duty cycle [%]
Halifax	14	0	42.2	5717.8	0.7
Rio	14	1	47.7	5712.3	0.8
Mumbai	14	1	39.2	5720.8	0.7

Table 3. 96-hour OGNS testing results summary: part 2. The average RMSE estimation accuracy for target flybys was 24.5 m and 0.40 m/s for position and velocity, respectively, which is comparable to the RDEKF algorithm with a 100% GPS duty cycle.

target	flyby duration [min]	3D position error		3D velocity error	
		RMSE [m]	max [m]	RMSE [m/s]	max [m/s]
Halifax	8.99	30.64	145.42	0.42	1.37
Rio	8.96	24.92	43.83	0.43	1.57
Mumbai	9.12	17.95	34.02	0.36	1.09

The average 3D RMSE for target flybys were 24.5 m and 0.40 m/s for position and velocity, respectively; these errors are similar in magnitude to those from the RDEKF algorithm with a 100% GPS duty cycle meeting the requirements for fine orbit determination (< 200 m estimation accuracy) as discussed previously in Section 2.3.3. Furthermore, the average GPS duty cycle was less than 1% for all three OGNS target tests with the GPS being turned on for less than an hour for the entire 4-day period.

These results demonstrate the OGNS algorithm's ability to automatically switch between ODS strategies to ensure fine estimation accuracy during target flybys while minimizing the use of onboard resources (i.e. GPS power consumption). The OGNS algorithm developed in this research can, and should, be configured to a specific satellite mission. The particular algorithm configuration implemented here is well-suited for CubeSat Earth observation missions with limited onboard resources; however, various combinations of ODS strategies, GPS duty cycle, and control logic may be selected to achieve different performance requirements. This concludes the research and development on the first thesis objective which is the orbital propagator.

In the following chapter, custom ADCS devices designed for the LORIS satellite are tested using the RSS platform.

Chapter 5. ADCS Device Testing

This chapter presents a summary of the ADCS device testing experiments performed using the RSS platform. Conducting system identification experiments to identify accurate sensor and actuator models will increase simulation accuracy. Furthermore, characterizing the performance of custom devices built for LORIS is imperative to evaluate their potential on future CubeSat missions. Finally, the tests described in this section provide an opportunity to evaluate the capabilities of the RSS platform as a tool to test ADCS devices and control strategies.

The first device tested in this research is a novel reaction wheel assembly (RWA) designed by the LORIS team [86]. Project timeline constraints prevented the use of the RWA onboard LORIS; however, a prototype was manufactured to experimentally validate the custom design. System identification experiments were performed to identify a motor model for the LORIS RWA. A slew controller was then designed for the RSS platform in simulation and tested experimentally to assess the validity of the identified motor model and evaluate the RWA performance.

The second device tested in this research is a coarse Sun sensor featuring a custom arrangement of COTS photodiodes. Sun sensors are used on LORIS in combination with a COTS magnetometer to estimate the satellite's attitude. To date, there has been no experimental verification of the Sun sensor used on LORIS. Furthermore, the ADCS simulator currently treats each photodiode equally, assuming constant light sensitivity; in practice however, each photodiode has a slightly different sensitivity which will affect estimation accuracy if unaccounted for. For these reasons, an experimental testbed is designed to test LORIS Sun sensor replicas with the RSS platform and a light source which emulates LEO sunlight conditions. Then, an experiment is performed to verify the Sun sensor's functionality and identify overall performance trends.

The chapter begins with a description of the RSS platform and other experimental apparatus used to test the LORIS RWA and Sun sensors. Then, the methodology and results of the RWA system identification, RSS slew controller, and Sun sensor experiments are discussed. Finally, conclusions are drawn regarding device performance, model accuracy, and the RSS platform utility for ADCS testing. Recommendations are made on improvements to the experimental setup and process.

5.1 RSS Platform

The RSS platform is a floating robot used to emulate the conditions of a microgravity environment in two-dimensions. The base of the RSS has planar air bearings which generate lift when pressurized. These air bearings, as well as onboard thrusters, actuate the RSS in the horizontal plane. With the addition of an onboard reaction wheel the RSS can also rotate about a single axis. Similar platforms have traditionally been used for on-Earth emulation of LEO proximity operations like active debris removal (ADR) and spacecraft rendezvous. One goal of this research was to show that a two-dimensional platform like the RSS may be useful to test ADCS equipment.

An annotated diagram of the RSS CAD assembly model with integrated ADCS equipment is shown in Figure 11.

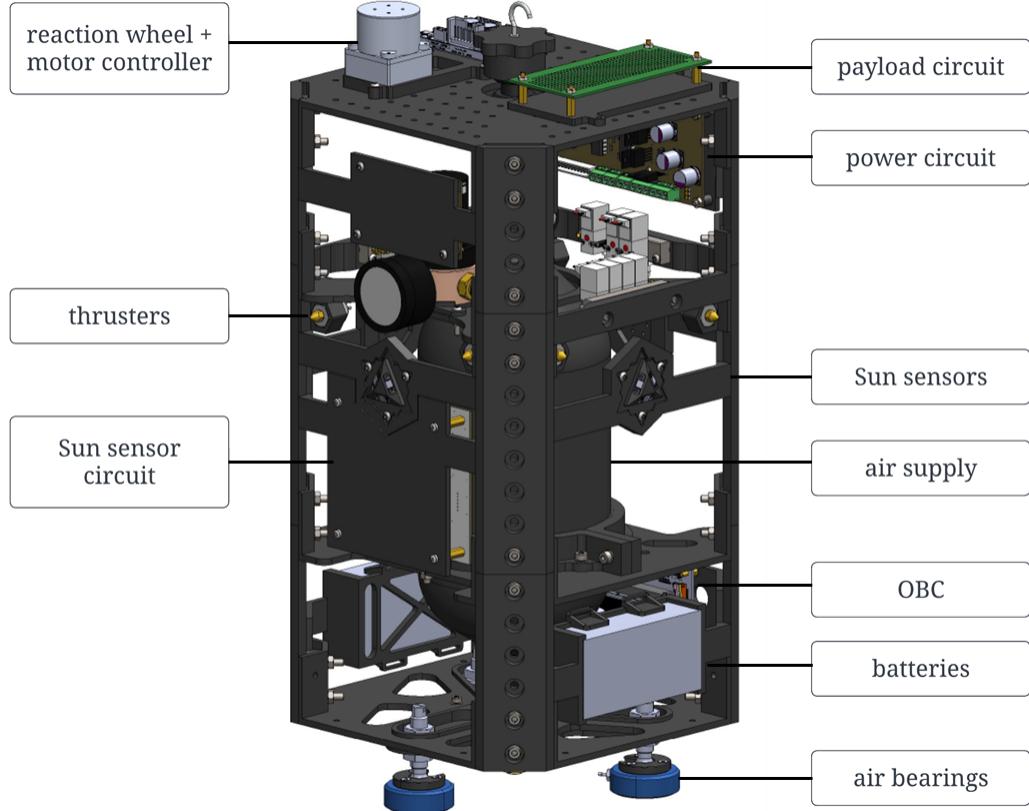


Figure 11. Robotic Spacecraft Simulator main subsystems/components.

An RSS system overview diagram is shown in Figure 12. The RSS onboard computer (OBC) uses the robot operating system (ROS) running in a Linux environment to send and receive commands, read sensor inputs, and control onboard actuators. The components related to the base operation of the RSS include the OBC, pneumatics,

batteries, and a custom power distribution circuit. Payload components include the reaction wheel and motor controller, Sun sensors and Sun sensor circuit, and the payload microcontroller circuit. In addition to management of onboard components, the RSS OBC also communicates wirelessly with the user workstation and the motion capture system (MOCAP) which estimates the RSS attitude in real-time to a high accuracy (~ 1 millimeter in linear displacement and 0.1° in angular displacement).

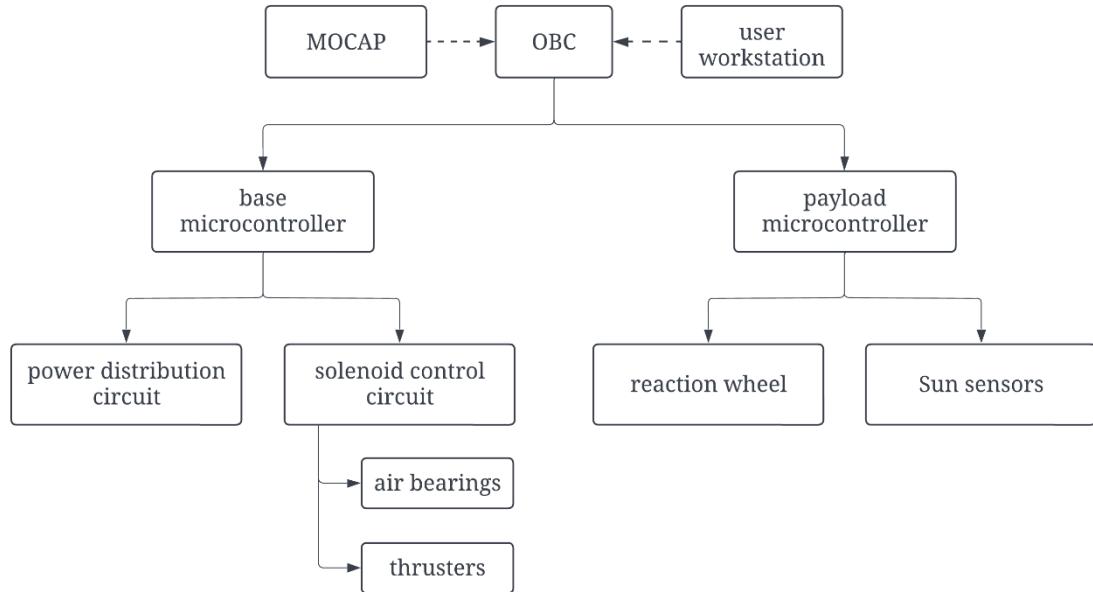


Figure 12. RSS system overview (NOTE: solid lines denote wired connections onboard the RSS platform, dashed lines denote wireless connections).

This chapter will focus on tests completed using the prototype RWA and Sun sensors; however, as these tests rely on other equipment from the RSS components diagram, some discussion of select components and their basic functions will be provided. To start, the MOCAP generates real-time estimates of RSS attitude as feedback to the RSS slew controller, discussed in Sections 5.3 and 5.4, and ground truth to evaluate ADCS performance. Operational commands for the RSS are provided by the user workstation. Command messages energize the air bearings air supply and generate the input definitions to actuate the RWA described in Sections 5.2 – 5.4. The payload microcontroller controls communication between the OBC and the payload devices. As previously discussed, the OBC is the centerpiece of the RSS platform during testing. It communicates with the onboard and external RSS components, time stamps test data, and implements control logic.

5.2 Reaction Wheel Testing

The goal of this section was to identify an accurate model for the LORIS RWA used in the ADCS simulator. To this end, two parametric motor model definitions are investigated to describe the RWA's speed response to current inputs. Response curves from experimental step input tests are used in conjunction with MATLAB's *fminsearch()* algorithm to identify parameters which best fit each motor model definition. Identified models are assessed based on their ability to reproduce step responses for a range of input magnitudes.

A secondary goal of this section was to evaluate the performance of the RWA and make recommendations for its use on future satellite missions. Additional step input tests are conducted to identify the RWA's available momentum storage and maximum continuous acceleration torque. These parameters are then compared to the original design requirements for the RWA.

Following this section, the best identified motor model for the RWA will be used in simulation to design a slew controller for the RSS platform. Comparisons between simulation and experiment is used to validate the identified RWA motor model for use in the ADCS simulator.

5.2.1 Reaction Wheel Motor Model Fundamentals

A reaction wheel can generally be described as an energy transducer which converts energy from the electrical to the mechanical domain. Figure 13 shows a simplified input-output model for a reaction wheel system.

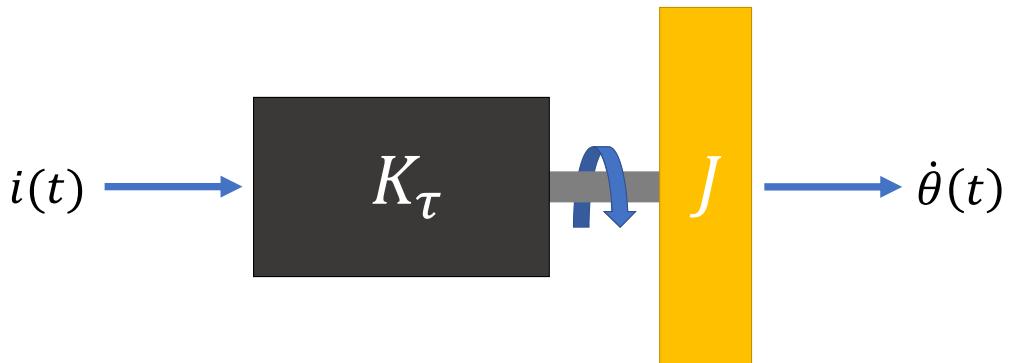


Figure 13. Reaction wheel system diagram modelled as a flywheel rigidly attached to a DC motor where $i(t)$ is the motor current, K_τ is the motor torque constant, J is the flywheel inertia, and $\dot{\theta}(t)$ is the angular speed of the flywheel.

In Figure 13, a DC motor is rigidly attached to a flywheel with inertia J . Input current to the motor $i(t)$ is shown to produce a resulting angular speed of the flywheel $\dot{\theta}$. The transformation between input and output for the reaction wheel system in Figure 13 can be described by two equations.

Starting with the electrical domain, DC motor torque τ_m can be related to motor armature current i_a by the following equation for a linear and idealized reaction wheel:

$$\tau_m = K_\tau i_a . \quad (5.2)$$

In Equation 5.2, the motor torque is directly proportional to the motor armature current through the motor torque constant K_τ .

The mechanical output of the flywheel can be related to the applied motor torque through Newton's Second Law of Motion. This law states that the acceleration of a body is dependent on two variables: the net force acting on the body and the object's mass. Equivalently, for a rotational system, the angular acceleration of a body is dependent on its net torque and rotational inertia. In addition to the motor torque, one must also account for frictional losses in the system. This leads to the following equation of motion:

$$J\ddot{\theta} = \tau_m - \tau_f . \quad (5.3)$$

In Equation 5.3, the angular acceleration $\ddot{\theta}$ of the flywheel of rotational inertia J is equal to motor torque τ_m subtracting the friction torque τ_f . Substituting Equation 5.2 into 5.3 results in the following basic motor model definition:

$$J\ddot{\theta} = K_\tau i_a - \tau_f . \quad (5.4)$$

The frictional reaction force is composed of multiple underlying mechanisms which are not easily captured by any one model; however, friction can largely be broken down into three observable mechanisms: static friction, coulomb friction, and viscous friction. Static friction is the breakaway force constant required for initial motion of a mass. Coulomb friction is the constant reaction force which resists motion and is smaller in magnitude than the initial static friction. Viscous friction is the component of friction assumed to increase linearly with speed.

The first motor model definition investigated only captures the effects of viscous friction. The ordinary differential equation which describes a viscous motor model is shown in Equation 5.5:

$$J\ddot{\theta} + b\dot{\theta} = K_\tau i_a . \quad (5.5)$$

Equation 5.5 models a forced second order linear-time-invariant system with damping. Viscous friction is modelled as the product of the viscous friction coefficient b and the flywheel angular speed $\dot{\theta}$. Equation 5.5 can be reduced to the first order differential equation shown in Equation 5.6:

$$\dot{y} + \frac{x_2}{x_1}y = \frac{x_3}{x_1}u \quad (5.6)$$

where,

$$y = \dot{\theta}$$

$$u = i_a$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} J \\ K_\tau \\ b \end{bmatrix}.$$

Applying the Laplace transform to both sides of Equation 5.6 and rearranging leads to the open-loop transfer function shown in Equation 5.7:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{x_3}{x_1 s + x_2} . \quad (5.7)$$

If viscous friction is the dominant mechanism in the friction torque term τ_f , then Equation 5.7 can be solved directly for y ; however, it is often the case that a more refined friction model is needed for a reaction wheel system.

Taking into account the combined effects of viscous and Coulomb friction results in the ordinary differential equation shown in Equation 5.8:

$$J\ddot{\theta} + b\dot{\theta} + c \operatorname{sgn}(\dot{\theta}) = K_\tau i_a . \quad (5.8)$$

As shown in Equation 5.8, Coulomb friction can be modelled as the product of the Coulomb friction coefficient c and the sign of angular speed $\operatorname{sgn}(\dot{\theta})$. The sign function introduces discontinuities for which there is no simple analytical solution. Equation 5.8 is

instead solved numerically using the first order Taylor series expansion shown in Equation 5.9:

$$y(t + dt) = y(t) + \dot{y}(t)dt \quad (5.9)$$

where,

$$y = \dot{\theta}$$

$$\dot{y} = \frac{1}{J} (K_\tau i_a - b\dot{\theta} - c \operatorname{sgn}(\dot{\theta})).$$

Equation 5.9 estimates the future state of the flywheel angular speed for sufficiently small time steps.

Moving forward, system identification experiments will be performed to identify parameters which best fit the viscous friction model in Equation 5.6 and the viscous-Coulomb friction model in Equation 5.8. The Taylor series expansion approximates both models in discrete form. The following subsection will discuss the procedure used to identify best fit parameters for each model using MATLAB.

5.2.2 Model Identification Procedure

MATLAB's *fminsearch()* function was used to identify parameters which best fit experimental measurements for the two motor model definitions in Section 5.2.1. The *fminsearch()* function is based on a simplex search which attempts to find a local minimum to a problem specified by $f(x)$ [87]. For the RWA motor model identification experiments, $f(x)$ is described by Equation 5.10:

$$f(x) = \left[\sum_{k=2}^N (y_k - \hat{y}_k)^2 \right]^{0.5} \quad (5.10)$$

$$\hat{y}_k = F(\boldsymbol{x}, \hat{y}_{k-1}, u_{k-1}, T_s) \quad (5.11)$$

where,

y_k = measured output for the current time step

\hat{y}_k = estimated output for the current time step

$F(\boldsymbol{x}, y_0, u_k, dt)$ = motor model definition

\boldsymbol{x} = model parameters

\hat{y}_{k-1} = estimated output for the previous time step

u_{k-1} = measured input for the previous time step

T_s = sample time .

Given a vector of initial parameter estimates \mathbf{x}_0 , the *fminsearch()* function independently varies model parameters to optimize the agreement between experimental measurements and the motor model. Poor initial parameter estimates, or inaccurate motor model definitions, may result in the algorithm failing to converge to a solution.

The *fminsearch()* algorithm may only converge to a *local* minimum for the described problem or cost function $f(x)$. In other words, the identified parameters may not be representative of their physical definitions – rather, the parameter vector \mathbf{x} only represents the model which best fits the experimental measurements used in the minimization process.

Using this methodology, parameters were identified using several different experimental datasets as inputs to the viscous friction and viscous-Coulomb friction motor model definitions. The test setup, experimental data, and results of the system identification tests will be described in the following subsections.

5.2.3 Reaction Wheel Test Setup

The LORIS RWA features a novel multi-shaft coupling system, dual bearing supported brass reaction wheel, and COTS Maxon EC20 flat motor. The electrical interface to the motor is made through a flexible PCB connector which is accessible outside of the RWA aluminum casing as shown in Figure 14.

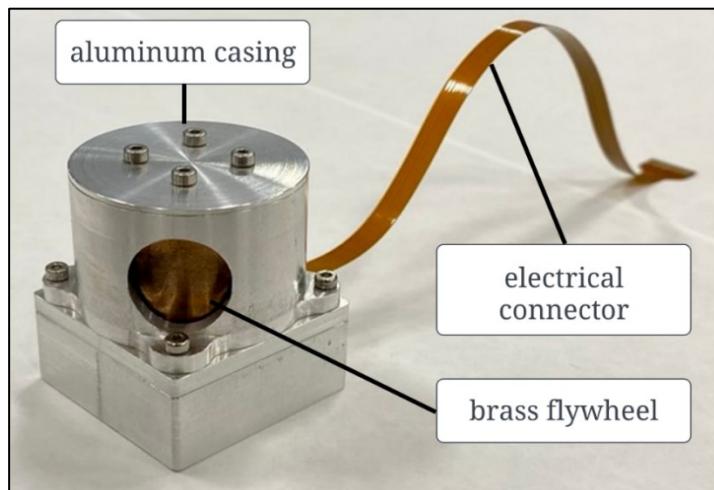


Figure 14. LORIS reaction wheel assembly used in this research.

The main design considerations for the RWA were based on the ADCS torque and momentum storage requirements for the LORIS satellite. Due to project timeline constraints, which prevented the use of the RWA onboard LORIS, only one prototype assembly was manufactured. In the interest of assessing the design for future CubeSat missions and improving the fidelity of the ADCS simulator, this RWA prototype was selected for testing.

Select properties for the LORIS RWA are provided in Table 4.

Table 4. Select LORIS RWA mechanical and electrical properties.

property	value
reaction wheel mass [kg]	7.486E-2
reaction wheel inertia [kg m ²]	1.011E-5
motor nominal current [A]	5.600E-1
motor stall current [A]	9.000E-1
motor torque constant [mNm A ⁻¹]	5.880E0

A COTS ESCON 36-3 motor controller from Maxon was selected for the RWA test setup used in this research based on its ease of integration and compatibility with the EC20 motor. The ESCON 36-3 controls the EC20 motor current based on an input pulse-width modulation signal from the payload microcontroller. Motor speed measurements were recorded directly from the motor controller.

5.2.4 Reaction Wheel Input Testing Summary

Step inputs were used to generate experimental responses for the system identification procedure described in Section 5.2.2. As shown in Table 4, the EC20 has a nominal current rating of 560 mA representing the maximum continuous current which can be applied to the motor without excessive overheating. Current inputs may exceed this rating up to a maximum of 900 mA for short durations; however, the motor controller will automatically reduce current based on the motor's thermal properties. To avoid controller interference, set point current commands were generated for step inputs ranging from -500 mA to 500 mA at 100 mA intervals.

For each step input test, the set point current and measured motor speed were time stamped and recorded. The raw sample time was approximately 39 ms; however,

measurements were resampled at a constant 10 ms time step using linear interpolation. The resulting speed response curves for positive current step input tests are shown in Figure 15.

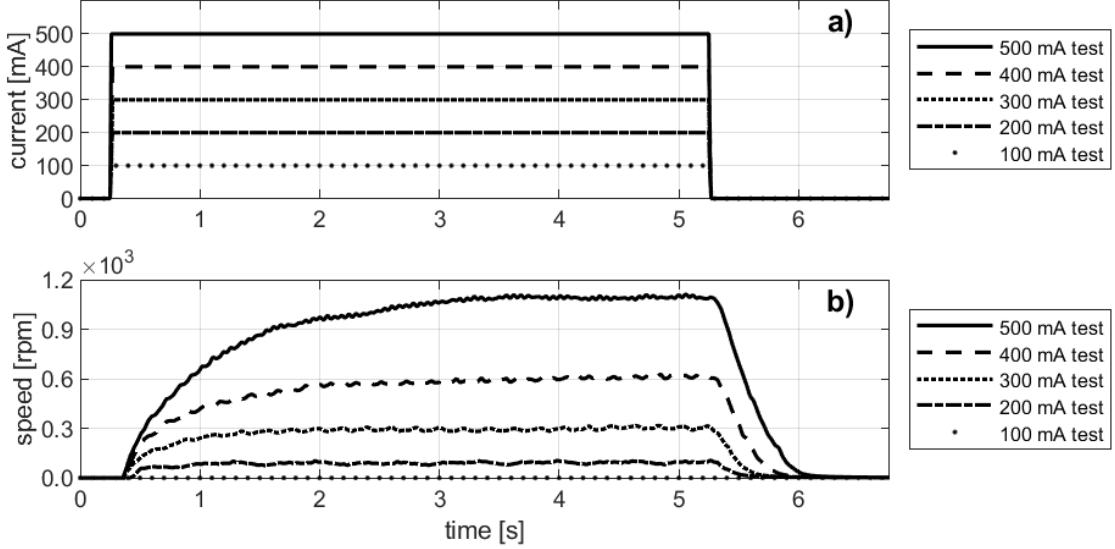


Figure 15. RWA positive step input test summary plots for set point currents within the motors nominal operating range: a) input current and b) output speed vs. time. The speed response is approximately first order suggesting significant damping in the system.

The set point current commanded by the RSS OBC is shown in Figure 15 a) and compared to the measured speed shown in Figure 15 b). Similar response curves for negative current inputs can also be seen in Figure 43 of Appendix C.3.1. No output response was observed for either the 100 mA or -100 mA input tests likely due to the presence of static friction in the system. The remaining speed curves are approximately first order which suggests significant damping of the output response.

In the following subsections, models are identified using the step input test responses for the viscous friction and viscous-Coulomb friction motor model definitions of Equation 5.6 and 5.8, respectively. Aside from the 100 mA input tests, experimental measurements from each test are used to identify a different parametrically defined motor model. Models are then evaluated based on their simulation accuracy for each step input test compared to the experimental measurements.

5.2.5 Viscous Friction Motor Model Identification

The viscous friction motor model definition was previously derived in Equation 5.5 and is repeated here for convenience:

$$J\ddot{\theta} + b\dot{\theta} = K_\tau i_a . \quad (5.5)$$

The discrete time solution to Equation 5.5 was solved numerically using the first order Taylor series expansion, as shown in Equation 5.12:

$$\hat{y}_k = \hat{y}_{k-1} + \frac{T_s}{x_1} (x_2 u_{k-1} - x_3 \hat{y}_{k-1}) \quad (5.12)$$

where,

$$\hat{y} = \dot{\theta}$$

$$u = i_a$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} J \\ K_\tau \\ b \end{bmatrix}.$$

As previously mentioned, the *fminsearch()* algorithm requires initial parameter estimates. Both J_0 and $K_{\tau 0}$ are based on predictions for the LORIS RWA. The flywheel inertia J_0 was calculated based on the material density and dimensions of the custom wheel design and is $1.011E - 5 \text{ kg m}^2$. The motor torque constant $K_{\tau 0}$ is directly from the EC20 datasheet and is $5.88E - 6 \text{ N m mA}^{-1}$. Finally, the initial estimate for the viscous friction coefficient b_0 is based on the findings of V. Carrara, 2010 [60] and is $5.16E - 6 \text{ N m s}$. The viscous friction coefficient from [60] was experimentally determined for a COTS reaction wheel design and is therefore different than that of the LORIS RWA; however, it is assumed that the friction parameters are close enough in magnitude to allow the *fminsearch()* algorithm to converge to a local minimum which more closely reflects the true RWA motor model.

The previously described step input responses were used to identify parameters which best fit the viscous friction motor model. Excluding the 100 mA and -100 mA step input tests, which result in zero speed output, each step input was used to identify a separate set of model parameters. The identified parameters for the viscous friction model are summarized in Table 5.

Table 5. RWA viscous friction motor model parameters identified from experimental step input tests using the *fminsearch()* algorithm.

step input [mA]	J [kg m ²]	K_τ [N m mA ⁻¹]	b [N m s]
500	7.400E-6	3.009E-6	1.385E-5
400	4.516E-6	1.617E-6	1.073E-5
300	4.209E-6	1.367E-6	1.351E-5
200	4.286E-6	7.026E-7	1.487E-5
-200	5.573E-6	1.474E-6	2.210E-5
-300	4.367E-6	1.558E-6	1.340E-5
-400	3.592E-6	1.412E-6	8.010E-6
-500	6.882E-6	2.894E-6	1.320E-5

Each of the parameter sets in Table 5 were used to simulate output responses across the entire range of step inputs and then compared to the experimental measurements. Simulation accuracy was evaluated with Equation 5.13:

$$\% \text{ fit} = 100 * \left(1 - \frac{[\sum_{k=1}^N (y_k - \hat{y}_k)^2]^{0.5}}{[\sum_{k=1}^N y_k^2]^{0.5}} \right). \quad (5.13)$$

In Equation 5.13, simulation error is normalized to the reference signal magnitude. This normalized error is then converted to a percent fit value to represent simulation accuracy. Normalizing error in this way allows for a more direct comparison of simulation accuracy for experimental responses of different magnitude, where higher percentages indicate better fits.

The resulting fit for each simulation model is summarized in Table 6.

Table 6. Goodness-of-fit for the identified viscous friction motor models (NOTE: Red cells indicate fits below 15 percent, yellow cells indicate fits above 15 percent and below 60 percent, and green cells indicate fits above 60 percent. The average goodness-of-fit for each model is also provided in the final column.)

step input test [mA]	+500	+400	+300	+200	-200	-300	-400	-500	avg.
+500 mA model [%fit]	89.3	56.2	< 0	< 0	< 0	15.9	75.0	88.5	40.6
+400 mA model [%fit]	68.2	90.3	51.7	< 0	< 0	69.6	81.9	67.4	53.6
+300 mA model [%fit]	46.3	66.3	91.7	< 0	48.6	83.9	56.7	45.8	54.9
+200 mA model [%fit]	21.8	31.3	46.4	89.7	69.4	40.3	26.7	21.6	43.4
-200 mA model [%fit]	30.7	44.1	65.4	57.2	91.4	56.8	37.7	30.4	51.7
-300 mA model [%fit]	53.1	75.6	83.1	< 0	26.6	90.2	64.8	52.5	55.7
-400 mA model [%fit]	78.8	80.7	27.4	< 0	< 0	49.0	88.9	77.8	50.3
-500 mA model [%fit]	89.2	54.9	< 0	< 0	< 0	14.2	74.0	88.6	40.1

The results show the models are more accurate when simulating the response of step inputs which are similar in magnitude to the set point current used in the system identification procedure. For example, the model identified using a positive 500 mA step input produces simulation accuracies above 85% when simulating the outputs of the positive and negative 500 mA step inputs; however, the simulation accuracy quickly drops off as this model is used to simulate the response of lower magnitude step inputs. To illustrate this, the simulated response of the model identified using the positive 500 mA step input is compared to experimental measurements for the positive 500 mA and positive 400 mA step input tests in Figure 16.

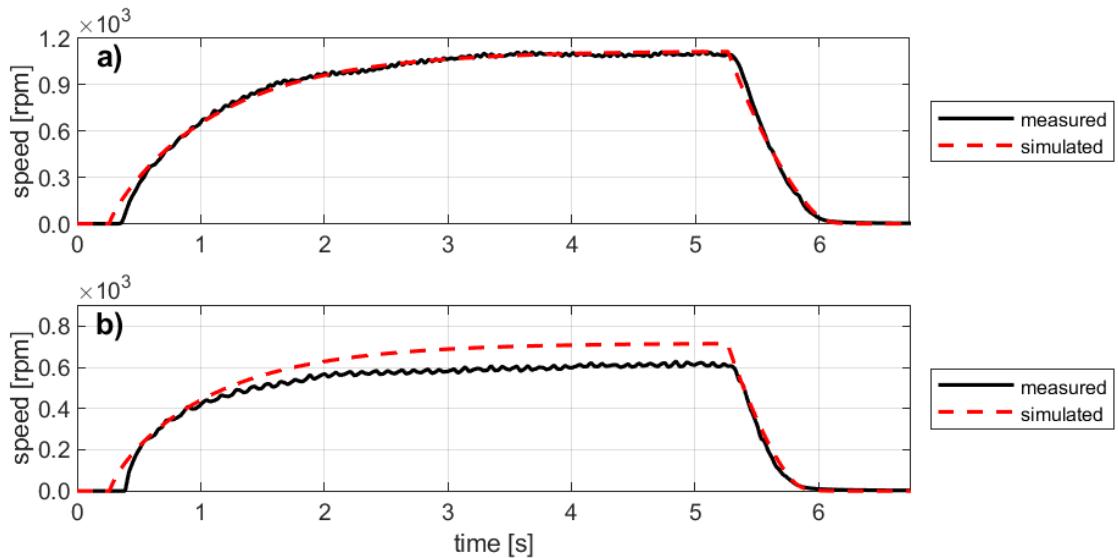


Figure 16. Simulated response of the viscous friction motor model identified using the positive 500 mA step input compared to experimental measurements: a) positive 500 mA step input test response and b) positive 400 mA step input test response vs. time. This model more accurately simulates output responses which are similar in magnitude to the set point current used in the system identification procedure.

Considering the poor average fit results shown in Table 6, it is likely that other dominant friction mechanisms are present in the RWA system. Therefore, in the next subsection, a viscous-Coulomb friction motor model will be tested using the same methodology.

5.2.6 Viscous-Coulomb Friction Motor Model Identification

The viscous-Coulomb friction motor model definition was previously derived in Equation 5.8 and is repeated here for convenience:

$$J\ddot{\theta} + b\dot{\theta} + c \operatorname{sgn}(\dot{\theta}) = K_\tau i_a . \quad (5.8)$$

High frequency oscillations were observed in the output response when Equation 5.8 was solved numerically. This can be attributed to discontinuities introduced by the *sgn()* function. For this reason, an approximation was made using the arctangent function as described in [88]. The updated viscous-Coulomb friction discrete motor model definition is shown in Equation 5.14:

$$J\ddot{\theta} + b\dot{\theta} + c \left(\frac{2}{\pi} \operatorname{atan} \left(\frac{\dot{\theta}}{\dot{\theta}_\delta} \right) \right) = K_\tau i_a . \quad (5.14)$$

In Equation 5.14, $\dot{\theta}_\delta$ represents a tuning parameter which defines the slope of the arctangent function. The smaller this value is, the more closely the continuous function represents the discontinuous sign function. The parameter $\dot{\theta}_\delta$ was tuned manually and set to a constant value of 8.

Solving Equation 5.14 numerically using the first order Taylor series expansion yields Equation 5.15:

$$\hat{y}_k = \hat{y}_{k-1} + \frac{T_s}{x_1} \left(x_2 u_{k-1} - x_3 \hat{y}_{k-1} - \frac{2x_4}{\pi} \operatorname{atan} \left(\frac{\hat{y}_{k-1}}{8} \right) \right) \quad (5.15)$$

where,

$$\hat{y} = \dot{\theta}$$

$$u = i_a$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} J \\ K_\tau \\ b \\ c \end{bmatrix} .$$

J_0 , $K_{\tau 0}$ and b_0 were initialized with the same parameters as those from Section 5.2.5. The initial estimate for c_0 is based on the findings of Carrara and Kuga, 2013 [61] and is $8.795E - 6 N m$.

The same methodology used for identifying the viscous friction model parameters (Section 5.2.5) was applied here. The identified parameters for the viscous-Coulomb friction model are summarized in Table 7.

Table 7. RWA viscous-Coulomb friction motor model parameters identified from experimental step input tests using the fminsearch() algorithm.

step input [mA]	J [kg m ²]	K_τ [N m mA ⁻¹]	b [N m s]	c [N m]
500	8.350E-6	3.826E-6	8.282E-6	9.831E-4
400	8.164E-6	3.229E-6	9.199E-6	7.632E-4
300	7.262E-6	2.607E-6	9.361E-6	5.793E-4
200	1.591E-5	2.721E-6	4.546E-6	8.966E-4
-200	1.217E-5	3.600E-6	5.685E-6	9.709E-4
-300	8.834E-6	3.503E-6	1.105E-5	7.524E-4
-400	8.034E-6	3.700E-6	7.521E-6	9.680E-4
-500	8.238E-6	3.897E-6	7.349E-6	1.108E-3

The resulting goodness-of-fit for each of the viscous-Coulomb friction simulation models is summarized in Table 8.

Table 8. Goodness-of-fit for the identified viscous-Coulomb friction motor models (NOTE: Red cells indicate fits below 15 percent, yellow cells indicate fits above 15 percent and below 60 percent, and green cells indicate fits above 60 percent. The average goodness-of-fit for each model is also provided in the final column.)

step input test [mA]	+500	+400	+300	+200	-200	-300	-400	-500	avg.
+500 mA model [%fit]	96.8	85.2	85.0	53.0	89.5	93.5	96.0	96.2	86.9
+400 mA model [%fit]	83.7	95.9	90.2	48.6	89.4	91.9	85.1	82.6	83.4
+300 mA model [%fit]	71.3	86.7	95.3	50.4	90.5	85.8	74.4	70.4	78.1
+200 mA model [%fit]	62.5	68.1	68.2	90.6	69.6	59.6	59.6	62.1	67.5
-200 mA model [%fit]	77.5	71.2	79.1	57.4	92.3	88.0	83.1	78.4	78.4
-300 mA model [%fit]	82.1	95.7	84.3	34.4	81.8	94.2	85.8	81.1	79.9
-400 mA model [%fit]	95.1	82.5	84.5	54.6	89.6	93.5	96.9	95.7	86.6
-500 mA model [%fit]	96.3	89.2	93.6	68.6	84.5	85.7	92.8	96.6	88.4

Similar to the viscous friction model, the results show that the simulation models are generally more accurate when reproducing the response of step inputs which are similar in magnitude to the set point current used to identify the model parameters. The viscous-Coulomb friction model does a better job simulating outputs across the entire operating range, aside from the 200 mA step input test which appears as an outlier compared to the other tests. To illustrate this, Figure 17 presents the measured speed response for the 500 mA and 400 mA tests compared to the simulation outputs for the viscous friction and viscous-Coulomb friction motor models identified using the 500 mA step input.

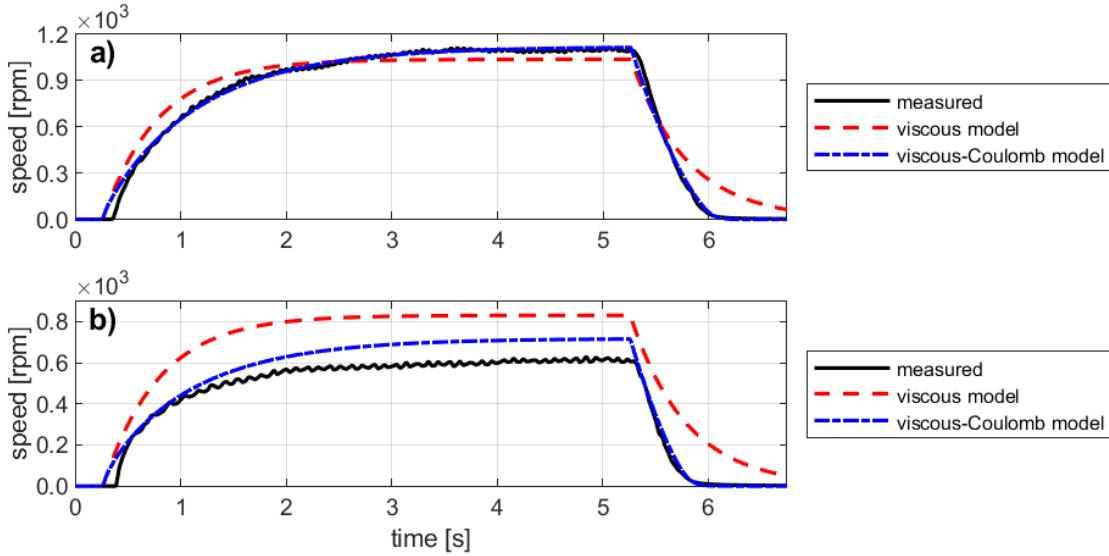


Figure 17. Simulated response of the viscous and viscous-Coulomb friction motor models identified using the positive 500 mA step input compared to experimental measurements: a) positive 500 mA step input test response and b) positive 400 mA step input test response vs. time. The viscous-Coulomb friction motor model more accurately simulates the output response for both tests.

As shown in Figure 17, the viscous-Coulomb friction motor model is able to simulate the RWA output more accurately than the viscous friction motor model for both the 500 mA and 400 mA step input tests.

The best performing viscous-Coulomb friction motor model was identified using a negative 500 mA step input and resulted in an average fit of 88.4%. Moving forward, this will be the selected RWA motor model.

In the following subsection, the LORIS RWA experimental performance is evaluated and compared to its original design requirements.

5.2.7 Reaction Wheel Performance Evaluation

The two primary metrics to evaluate the RWA's performance are its available momentum storage and maximum continuous acceleration torque. For the LORIS satellite, these metrics are based on requirements for both environmental disturbance torque rejection and attitude control.

The available momentum storage for the RWA is determined from its maximum speed range ω_{max} and inertia I_{RW} . The maximum RWA speeds recorded for each of the step input tests conducted in Section 5.2.4 are summarized in Table 9.

Table 9. RWA step input test summary based on the step input tests from Section 5.2.4.

set current [mA]	500	400	300	200	100	-100	-200	-300	-400	-500
max speed [rpm]	1115	629	318	106	0	0	-142	-358	-745	-1179

As shown in Table 9, the RWA achieves a maximum speed of 1115 rpm in the positive direction and a maximum speed of 1179 rpm in the negative direction. Taking the lesser of these two values, the maximum momentum storage L_{RW} for the RWA, when starting from rest, is calculated as follows:

$$L_{RW} = I_{RW}\omega_{max} = 1.181E - 3 \text{ N m s}$$

where,

$$I_{RW} = 1.011E - 5 \text{ kg m}^2$$

$$\omega_{max} = 1115 \text{ rpm} = 116.8 \text{ rad s}^{-1}.$$

The original RWA momentum storage design requirement for the LORIS satellite was 1.27E-3 N m s based on the expected environmental disturbance torque accumulation over one orbit [86]. Even without a safety factor, the available momentum storage of the RWA is unable to meet this requirement.

This discrepancy may indicate the significant presence of friction in the RWA system. In future RWA designs, it is recommended that more consideration be given to tolerancing and bearing selection to minimize friction in the system. Alternatively, a larger motor would increase the maximum achievable speed.

To optimize the available momentum storage for the RSS slew controller tests, current inputs outside of the EC20 motor's nominal range were considered. As previously discussed, the EC20 motor can generate currents up to 900 mA for short durations (less than 10 seconds depending on input magnitude). To identify the absolute maximum steady-state speed of the RWA, additional tests were conducted with positive and negative step inputs between 600 and 800 mA. The resulting speed responses are shown in Figure 18.

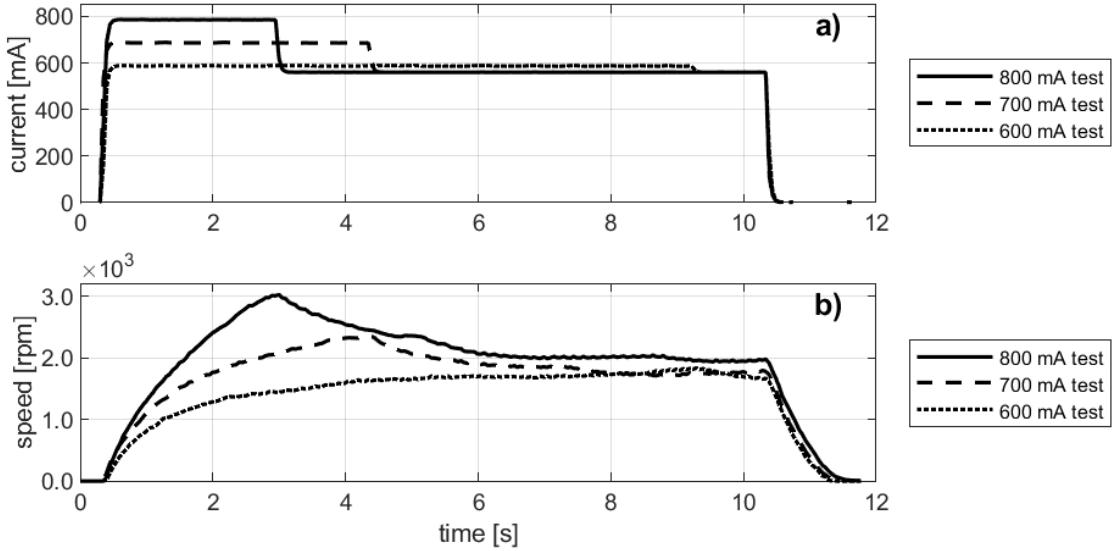


Figure 18. RWA positive step input test summary plots for set point currents outside the motors nominal operating range: a) input current and b) output speed vs. time. The maximum steady-state speed reached by the RWA is increased with set point currents above the motors nominal operating range; however, the motor controller automatically limits the applied current after a set time period to prevent overheating.

As shown in Figure 18, motor current is automatically limited by the ESCON 36-3 motor controller after a time period which is determined internally by the controller. The steady-state current for each step input test is 560 mA. After the current drops, the output response levels out to a steady-state speed. A similar output response plot for the negative step inputs is shown in Figure 44 of Appendix C.3.1. The results of the maximum step input tests are summarized in Table 10.

Table 10. RWA step input test summary based on the step inputs from Figure 18. The RWA is able to reach higher speeds when using set point currents above the motors nominal operating range.

set point current [mA]	800	700	600	-600	-700	-800
steady-state current [mA]	560	560	560	-560	-560	-560
maximum speed [rpm]	3021	2346	1833	-2076	-2679	-2950
steady-state speed [rpm]	1956	1791	1670	-2019	-2080	-2170

As indicated by the maximum speed results in Table 10, the RWA is able to reach significantly higher speeds when using current inputs above the motors nominal operating range; however, because these current inputs can only be maintained briefly, the steady-state measurements define the working speed range. A mid-range value of 1800 rpm was selected as the maximum speed moving forward. This results in a final momentum storage of 1.906E-3 N m s.

The maximum continuous acceleration torque metric is the effective torque the RWA is able to maintain during acceleration up to its maximum speed. This effective torque is not to be confused with the applied motor torque. Instead, it is equal to the applied motor torque minus the friction losses for any given speed. This was previously shown on the right-hand side of Equation 5.3, which is reproduced here for convenience:

$$J\ddot{\theta} = \tau_m - \tau_f . \quad (5.3)$$

Equation 5.3 can be used with speed measurements from the experiments to calculate the effective motor torque as a function of speed. A finite difference approach was used to estimate the motor acceleration $\ddot{\theta}$ centered around the maximum steady-state speed of 1800 rpm for the positive and negative 800 mA step input tests. The justification for this approach, compared to more rigorous numerical integration techniques, is that acceleration is approximately constant over small speed ranges. To illustrate this, a comparison of the measured speed and approximate constant acceleration speed is shown for the positive 800 mA step input test in Figure 19.

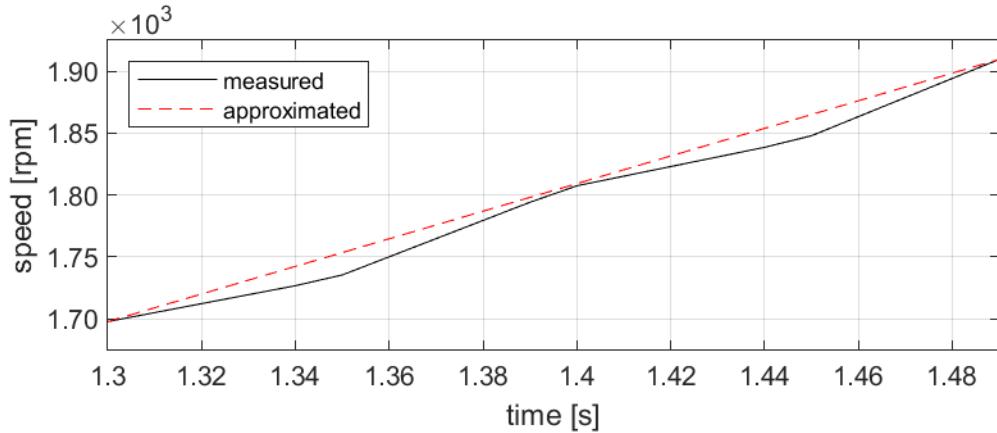


Figure 19. RWA constant acceleration speed response compared to experimental measurements for the positive 800 mA step input test. As expected, the constant acceleration approximation closely matches the experimental response over small speed ranges.

In Figure 19, the approximate speed is calculated based on the average acceleration between 1700 and 1900 rpm. This approximated speed closely matches the experimental speed with a maximum error of 18.1 rpm. Similar results are shown for the negative 800 mA step input test in Figure 45 of Appendix C.3.1. Using Equation 5.3 with the smaller of the two constant acceleration values results in a maximum continuous acceleration torque estimate of 1.157E-3 N m.

5.3 RSS Slew Controller Design

Increasing the fidelity of the LORIS ADCS simulator sensor and actuator models will lead to a better performing attitude control system (ACS); however, it is infeasible to design a simulation environment that is 100% accurate. For this reason, it is not advised to validate control strategies based on simulation results alone.

The primary goals of this section were to demonstrate the ability of the RSS platform to test ACS strategies and to validate the RWA motor model identified previously. To this end, a slew controller was initially designed in simulation with an RSS attitude control simulator then tested experimentally using the RSS experimental testbed. The simulated performance is compared to what was observed experimentally then comments made on the model accuracy.

5.3.1 RSS Attitude Control Simulator

The RSS attitude control simulator was developed in MATLAB/Simulink and uses the Simscape multibody toolbox to simulate the RSS rigid-body dynamics. A simplified assembly model for the RSS platform consisting of the RSS body frame and RWA flywheel was generated in Solidworks and imported into Simulink to represent the physical system.

Mass and inertial properties for the simplified flywheel were modified to reflect the actual RWA flywheel design. Mass and inertial properties for the simplified RSS body frame were modified based on estimates from the RSS detailed assembly model. In Simscape, the RW is mechanically integrated to the RSS body via a one-dimensional revolute joint. The RSS body is then connected to the world frame via a two-dimensional planar joint. Finally, the RW speed is controlled via a Simulink block representing the previously identified RW motor model.

With this setup, torques generated during acceleration of the flywheel are imparted onto the RSS platform which is then free to rotate in the world frame. The open-loop system diagram which describes the RSS Simulink model is shown in Figure 20.

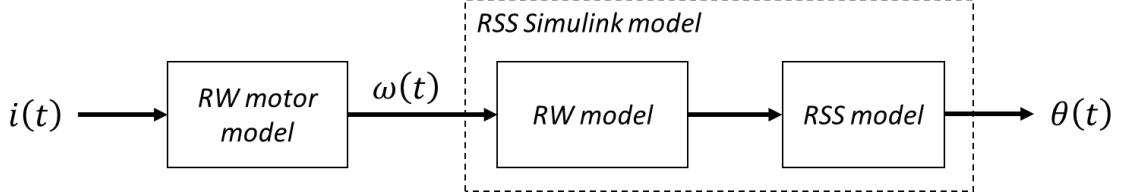


Figure 20. RSS open-loop system for the RSS attitude control simulator developed in MATLAB/Simulink where $i(t)$ is the set point current, $\omega(t)$ is RWA speed response, and $\theta(t)$ is the RSS slew angle. This is the open-loop system for the RSS slew controller in Figure 21.

The open-loop system of Figure 20 shows the input-output relationship between the RW set point current and the RSS slew angle assuming no external perturbations act on the system. From this relationship, a closed-loop slew controller was developed and implemented in Simulink as shown in Figure 21.

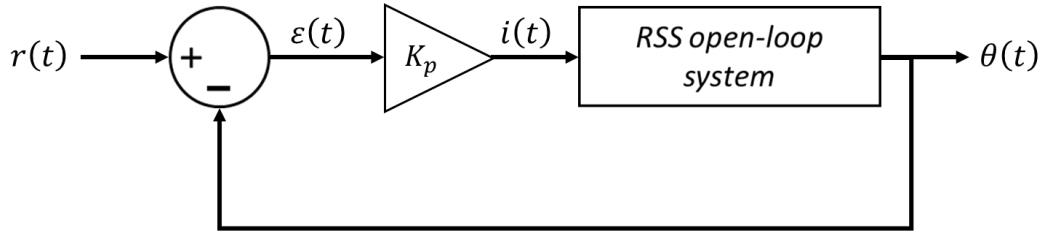


Figure 21. RSS closed-loop slew controller. The proportional gain term K_p is multiplied by the RSS slew error $\varepsilon(t)$ to generate the set point current $i(t)$ which is supplied to the RSS open-loop system in Figure 20.

The diagram shown in Figure 21 describes a proportional feedback controller which generates a RW current input as a function of slew error $\varepsilon(t)$ multiplied by gain K_p . The slew error is equal to the error between the reference $r(t)$ and measured slew angle $\theta(t)$ at time t .

In the following subsection, the RSS slew trajectory which defines the reference slew angle at each time step is described.

5.3.2 RSS Slew Trajectory Definition

The trajectory to test the RSS slew controller is composed of both positive and negative slew maneuvers. The trajectory definition begins with a positive slew maneuver defined by the set point angle θ_{slew} . The resulting final slew angle is maintained briefly and then followed by a negative slew maneuver returning the RSS to its original orientation. This trajectory definition was selected to test the controller's ability to perform positive slews, negative slews, and station-keep.

Slew maneuvers were generated using constant positive acceleration for half of the slew period and constant negative acceleration for the second half of the slew period, as defined by Equation 5.16:

$$\begin{bmatrix} r(t) \\ \dot{r}(t) \\ \ddot{r}(t) \end{bmatrix} = \begin{bmatrix} 1 & dt & 0.5 * dt^2 \\ 0 & 1 & dt \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r(t - dt) \\ \dot{r}(t - dt) \\ \alpha \end{bmatrix} \quad (5.16)$$

where,

$$\alpha = \begin{cases} 0, & t = 0 \\ \alpha_{slew}, & 0 < t \geq \frac{T}{2} \\ -\alpha_{slew}, & \frac{T}{2} < t \geq T \end{cases}$$

In Equation 5.16, $r(t)$ is the reference slew angle, dt is the sample time, T is the slew maneuver duration, and α_{slew} is the set point slew acceleration. Slew maneuvers which can be tested on the RSS are limited by the capabilities of the RWA. While there are no direct limitations of the set point slew angle, the maximum slew rate $\omega_{slew,max}$ and maximum slew acceleration $\alpha_{slew,max}$ are dependent on the RWA available momentum storage L_{RW} and maximum continuous acceleration torque τ_{max} , respectively.

Based on the RWA performance metrics derived in Section 5.2.7, the maximum slew rate and slew acceleration for the RSS can be calculated as follows:

$$\omega_{slew,max} = \frac{L_{RW}}{I_{RSS}} = \frac{1.906E - 3 \text{ N m s}}{2.605E - 2 \text{ kg m}^2} \frac{180 \text{ deg}}{\pi \text{ rad}} = 4.19 \text{ deg s}^{-1} \quad (5.17)$$

$$\alpha_{slew,max} = \frac{\tau_{max}}{I_{RSS}} = \frac{1.157E - 3 \text{ N m}}{2.605E - 2 \text{ kg m}^2} \frac{180 \text{ deg}}{\pi \text{ rad}} = 2.54 \text{ deg s}^{-2}. \quad (5.18)$$

As shown in Equations 5.17 and 5.18, the RWA can accelerate the RSS at 2.54 deg s^{-2} up to a slew rate of 4.19 deg s^{-1} before it reaches a maximum speed of 1800 rpm.

These slew trajectory limits assume there are no external torques acting on the RSS. For tests completed in simulation, this is true; in practice, however, there are external disturbances. For these reasons, only short slew maneuvers with relatively low speeds and accelerations were considered for the initial experimental validation tests completed in this research.

The final RSS slew trajectory was generated with Equation 5.16 based on 20-degree there-and-back slew maneuvers with a set point slew acceleration of 0.100 deg/s^2 . The resulting slew, slew rate, and slew acceleration for the 20-degree slew trajectory definition are shown in Figure 22.

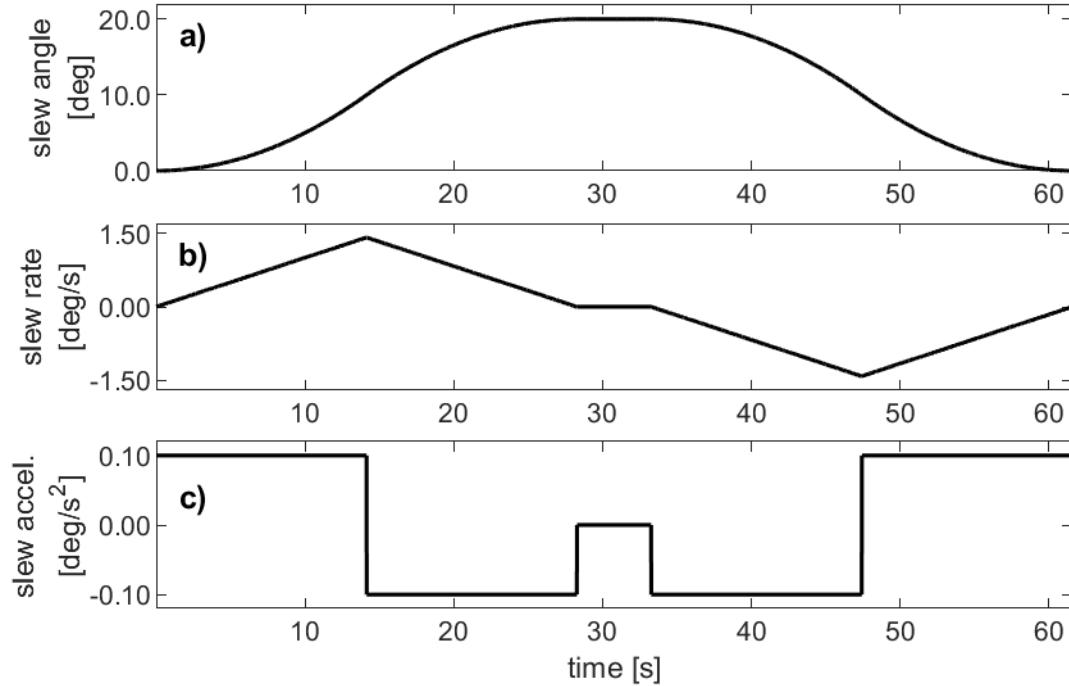


Figure 22. 20-degree slew trajectory for testing the RSS slew controller: a) slew angle, b) slew rate, and c) slew acceleration vs. time.

As shown in Figure 22, the maximum slew rate during the 20-degree slew trajectory is 1.41 deg/s . By operating at slew rates below the maximum theoretical value calculated in Equation 5.17 (4.19 deg/s), the RWA can compensate for small disturbance torques which may be present in the experimental test setup.

In the following subsection, the procedure and results for a manual controller tuning campaign completed using the RSS attitude simulator are described.

5.3.3 RSS Slew Controller Simulation Tuning

A proportional gain slew controller was designed for the RSS platform based on the 20-degree there-and-back slew trajectory from Section 5.3.2. In this subsection, the RSS attitude control simulator simulates the RSS closed-loop dynamics to tune the proportional gain K_p .

The evaluation criteria for tuning the proportional gain was based on the accuracy and stability of the controller. Both the absolute maximum slew error and average slew error could be used to evaluate controller accuracy; however, the absolute maximum error was used as the primary metric. To evaluate stability, a moving standard deviation for the control input was calculated and the maximum instantaneous deviation was used for comparison between controllers.

A manual strategy was used to tune the RSS slew controller. Starting with unity, the proportional gain was slowly increased in magnitude until the loss of controller stability which was identified by a significant increase in the maximum instantaneous deviation for the control input. The best performing controller from this was selected for experimental testing. The tuning results are summarized in Tables 11 and 12.

Table 11. RSS slew controller simulation tuning, part 1: summarized results of the tracked performance metrics for the proportional slew controller tests. The best performance is observed for test number 4; increasing the proportional gain further results in a loss of controller stability as indicated by the significant increase in the maximum current instantaneous standard deviation.

test	K_p [mA/deg]	stable	max current std. [mA]	max error [deg]
1	1E0	yes	0.05	29.4
2	1E1	yes	0.53	23.8
3	1E2	yes	1.58	4.14
4	1E3	yes	2.43	0.421
5	5E3	no	151.36	0.443

Table 12. RSS slew controller simulation tuning, part 2: tracked performance metrics for the proportional slew controller tests.

test	max current [mA]	max current std. [mA]	max speed [rpm]	max error [deg]	average error [deg]
1	-29.4	0.05	-14.2	29.4	1.478E1
2	-238.2	0.53	-191.0	23.8	8.824E0
3	-414.3	1.58	-751.3	4.14	1.546E-1
4	421.1	2.43	766.9	0.421	1.783E-3
5	-850.0	151.36	-1425.2	0.443	4.427E-4

As shown in Tables 11 and 12, large slew errors manifest in simulation until the proportional gain term is sufficiently large in magnitude. Test number four used a proportional gain equal to 1000 mA/deg and resulted in a maximum slew error of 0.421 degrees representing the best simulation accuracy achieved during testing. In test number

five, the proportional gain was increased to 5000 mA/deg and resulted in a maximum instantaneous current standard deviation which is orders of magnitude larger than that of the previous tests, indicating a loss of controller stability. The average error for test five was less than half that of test four which suggests that additional tuning may lead to better performance; however, the performance achieved in test four was considered sufficient for the experimental validation tests described in the following subsections. Result summary plots for tests four and five can be found in Figures 46 and 47 of Appendix C.3.2, respectively.

In the following subsection, the test setup used for the RSS slew controller experiments performed in Section 5.3.5 is introduced.

5.3.4 RSS Slew Controller Test Setup

As previously discussed, the RSS platform can be used to simulate a microgravity environment in two-dimensions. Three pressurized COTS air bearings at the base of the RSS generate a small film of air which lifts the platform allowing for near-frictionless motion. The film thickness is on the order of 5 microns [89]. Because the film thickness is so small, the RSS is sensitive to imperfections in the surface it is floating over. Additionally, imbalances of the RSS center of mass over the air bearings as well as surface level gradients cause the platform to drift while the air bearings are active.

For the RSS slew controller tests, a solution was required to reduce platform drift while minimizing the impact of external disturbance torques. One possible solution was to use the onboard thrusters to station-keep; however, while this may eliminate the drift issue, the thrusters may introduce additional disturbance torques to the system. As an alternative solution, a fishing line was tethered to the RSS from above to keep it in place. By attaching a line in tension from above the RSS along its axis of rotation, drift could be reduced with minimal impact on the platform's rotational dynamics.

The final test setup consisted of a granite surface plate with leveling jacks, the RSS with an attachment point for the fishing line, and an overhead stand which was the secondary attachment point for the fishing line. An annotated diagram of the RSS test setup is shown in Figure 23.

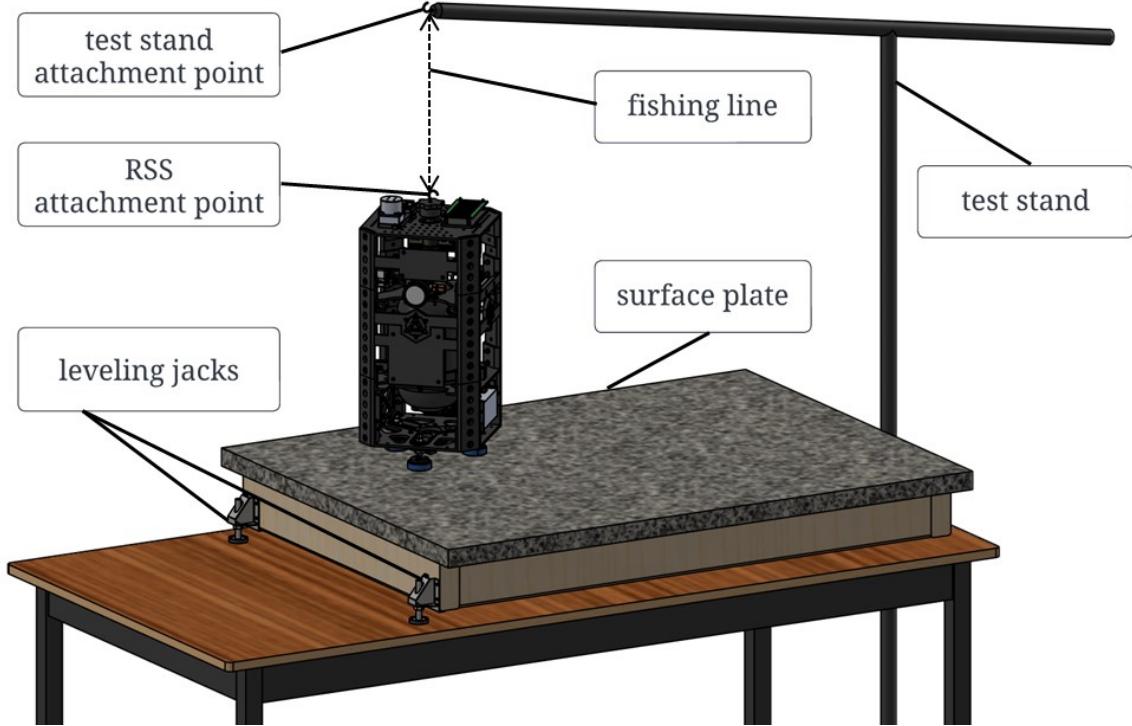


Figure 23. RSS slew controller experimental testbed, including RSS with overhead fishing line attachment, test stand, and granite platform with leveling jacks.

Tension was added to the fishing line to prevent significant translational drift while the RSS was supported by air bearings. The gauge of the fishing line was sufficiently small as to minimize torsion effects induced during rotation of the RSS platform.

In the following subsection, the RSS slew controller which was designed in simulation (Section 5.3.3) is validated experimentally using the RSS experimental testbed.

5.3.5 RSS Slew Controller Experiments

The RSS slew controller designed using the RSS attitude control simulator from Section 5.3.3 was validated experimentally using the same 20-degree there-and-back slew trajectory from Section 5.3.2. Ground truth attitude measurements were transmitted to the RSS OBC wirelessly from the Raptor 4S motion capture system [90]. MOCAP measurements provided real-time controller feedback as well as ground truth attitude reference to evaluate the RSS slew controller performance.

Summary plots comparing the 20-degree RSS slew controller simulation performance to experimental measurements are shown in Figure 24.

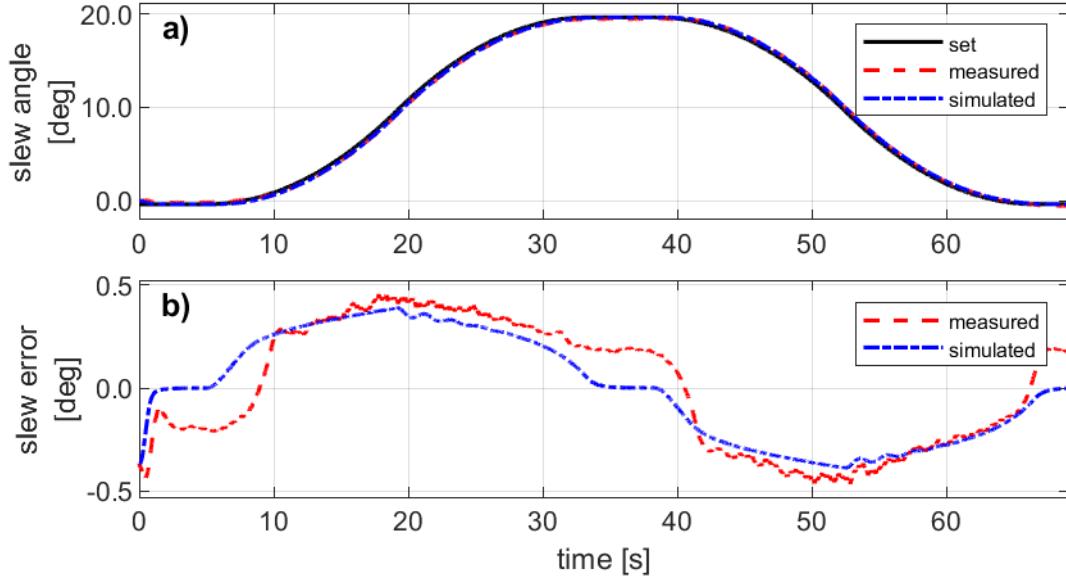


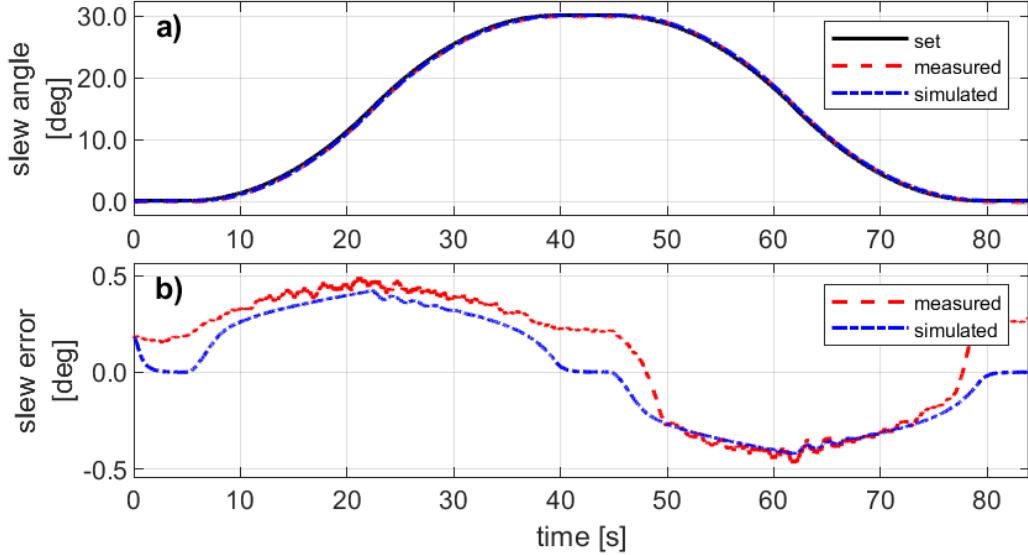
Figure 24. RSS slew controller performance summary for a 20-degree there-and-back slew trajectory test:
a) RSS slew angle and b) RSS slew error vs. time. Note the good agreement between the simulated and experimental results.

In Figure 24 a), the set point trajectory is plotted against the experimentally measured and simulated slew response. The slew error for both the measured and simulated response is shown in Figure 24 b). Simulation accuracy was once again calculated using Equation 5.13 and resulted in a 99.0 percent fit between the simulated and measured response.

During the positive slew maneuver, the relative slew error was positive; the opposite was seen for the negative slew maneuver. This result shows that the system response lags behind the set point trajectory and is indicative of the limitations of the proportional control scheme. One noticeable difference in the measured response compared to the simulated response was the steady-state error observed while station-keeping. This difference may be a result of the small disturbance torques which are present in the experimental setup.

An additional test was performed for a 30-degree there-and-back slew trajectory. This slew trajectory was generated using the same methodology as that of the 20-degree slew trajectory discussed in Section 5.3.2. The maximum slew acceleration was kept at 0.100 deg s^{-2} which resulted in an increased maximum slew rate of 1.732 deg s^{-1} . The 30-degree set point trajectory definition is shown in Figure 48 of Appendix C.3.2.

Summary plots comparing the 30-degree RSS slew controller simulation performance to experiment are shown in Figure 25.



*Figure 25. RSS slew controller performance summary for a 30-degree there-and-back slew trajectory test:
a) RSS slew angle and b) RSS slew error vs. time. These results further demonstrate the good agreement
between the RSS simulation model and the RSS experimental testbed.*

The results for the 30-degree slew trajectory test are similar to those of the 20-degree slew trajectory test; however, the simulation fit was slightly better at 99.3 percent.

Both the 20-degree and 30-degree tests show excellent agreement between the simulation model and the experimental measurements. A proportional gain controller tuned using the RSS attitude controller simulator was successfully able to control the RSS's attitude to set point trajectories. The maximum pointing error observed during testing was under 0.5 degrees which is well below the LORIS mission requirement of an attitude pointing accuracy of ± 10 degrees [1]. Improved performance can be expected through additional tuning of the proportional controller or by implementing more advanced control strategies.

Both goals outlined at the beginning of this section have been achieved. The RSS platform was successfully used to test ACS devices and control strategies which lead to conclusions on their performance. Additionally, the agreement between predictions from simulation and experimental measurements validate the accuracy of the identified RWA motor model.

In the following section, similar experiments will be run using the RSS platform to test custom Sun sensors designed by the LORIS CubeSat team.

5.4 Sun Sensor Testing

The Sun sensor experiments completed in this research are intended to demonstrate the capabilities of the RSS experimental testbed for the attitude determination system (ADS) sensor testing. Using the test setup described in Section 5.4.3, the RSS platform was manually actuated to an arbitrary slew rate and left spinning while recording light intensity measurements from onboard photodiodes. The recorded light intensity measurements were post-processed as described in Section 5.4.1. From the raw and post-processed photodiode intensity measurements, conclusions are drawn regarding the advantages and disadvantages of the test setup, and recommendations for future experiments.

5.4.1 Sun Vector Estimation Theory

Photodiodes are sensors which output current as a function of intensity and relative orientation of a light source. The measured light intensity of a single photodiode can be described by Equation 5.19 [91]:

$$I = I_0 \cos \theta + \eta . \quad (5.19)$$

In Equation 5.19, the measured intensity I is calculated as the product of the light source intensity I_0 multiplied by the cosine of the angle θ with additive measurement noise η . The angle θ is the orientation of the photodiode normal vector \mathbf{n} to the light source direction vector \mathbf{s} in a plane. The definition of the variables from Equation 5.19 are shown in Figure 26.

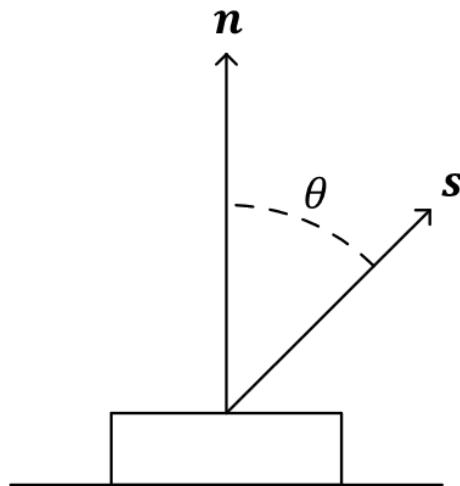


Figure 26. Photodiode light vector geometry.

If the light source intensity is known, the relative intensity I/I_0 can estimate the cosine angle between the photodiode normal and the light source direction vectors. This relationship forms the basis for Sun vector estimation.

At least three non-parallel photodiodes must be used to estimate a Sun vector in three dimensions. In [91], Equation 5.20 is derived to relate relative light intensity readings from multiple photodiodes to the incident Sun vector \mathbf{s} :

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \boldsymbol{\eta} \quad (5.20)$$

where,

$$\mathbf{y} = \begin{bmatrix} I_1 \\ I_{0,1} & I_2 \\ & I_{0,2} & \cdots & I_k \\ & I_{0,k} \end{bmatrix}^T$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \\ \vdots \\ \mathbf{n}_k \end{bmatrix} = \begin{bmatrix} n_{1,x} & n_{1,y} & n_{1,z} \\ n_{2,x} & n_{2,y} & n_{2,z} \\ \vdots & \vdots & \vdots \\ n_{k,x} & n_{k,y} & n_{k,z} \end{bmatrix}$$

$$\boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ I_{0,1} & \eta_2 \\ & I_{0,2} & \cdots & \eta_k \\ & I_{0,k} \end{bmatrix}^T.$$

The Sun vector in Equation 5.20 is specified in the BF coordinate system and is related to the relative intensity vector \mathbf{y} by the observation matrix \mathbf{H} , where the observation matrix is composed of photodiode normal vectors \mathbf{n}_i . The relative intensity for photodiode i is calculated as the quotient of the measured intensity I_i and the scale factor $I_{0,i}$, where the scale factor is an experimentally determined constant used to normalize measurements from multiple photodiodes with differing light sensitivities.

Solving Equation 5.20 for the estimated Sun vector $\hat{\mathbf{s}}$ results in the following equation [91]:

$$\hat{\mathbf{s}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y} \quad (5.21)$$

where,

$$\hat{\mathbf{s}} = [\hat{s}_x \quad \hat{s}_y \quad \hat{s}_z]^T.$$

Equation 5.21 estimates the three-dimensional Sun vector in the BF coordinate system as long as a minimum of three non-parallel photodiode intensity measurements are available. However, there are several factors which may affect the accuracy of the Sun vector estimate.

To start, there is measurement noise associated with each sensor. This noise may be combated by using a weighted least square solution [91]. Additionally, photodiodes will have differing light sensitivities which may skew the resulting Sun vector estimate if unaccounted for. Characterizing the sensitivity of photodiodes and adjusting light intensity measurements by appropriate scale factors minimizes these impacts; however, it has been shown that photodiodes subjected to the harsh conditions of LEO will also degrade with time leading to decreased sensitivities. In [91], an on-orbit photodiode calibration technique is presented to address photodiode sensitivity degradation. Finally, in LEO one must also account for the effects of Earth albedo and possibly other light sources depending on the mission accuracy requirements. One possible solution to eliminate the impacts of additional light sources is to set a minimum intensity threshold for measurements to estimate the Sun vector in Equation 5.21.

In this research, the RSS platform is used to test custom photodiode arrangements designed for the LORIS satellite. To this end, the Sun vector estimate from Equation 5.21 is transformed into a two-dimensional slew angle estimate and compared to its ground truth. The Sun vector estimate in the RSS BF as well as the estimated slew angle are shown in Figure 27.

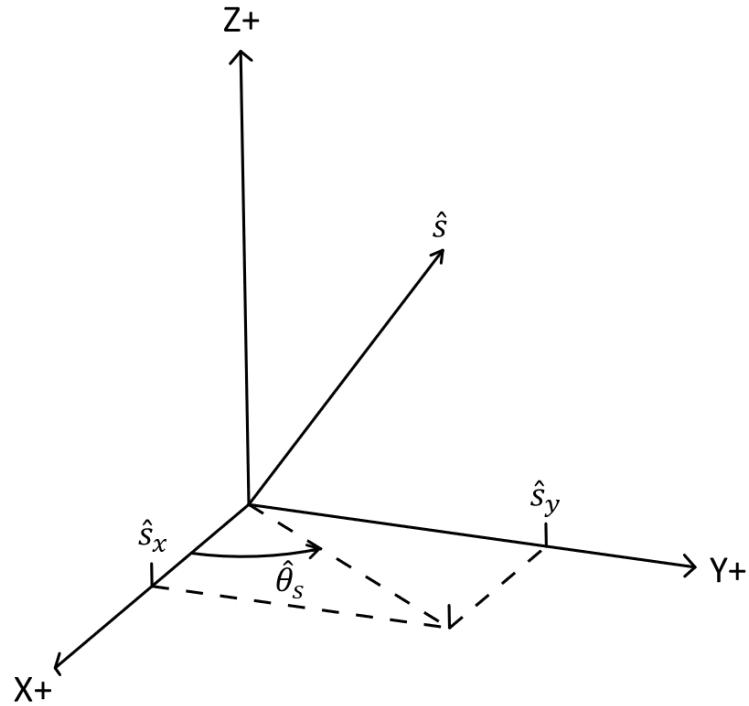


Figure 27. 3D Sun vector estimate parameters of Equation 5.21.

In Figure 27, the estimated Sun vector is shown along with its 2D projection onto the RSS BF xy-plane. The slew angle estimate $\hat{\theta}_s$ is shown as the counter-clockwise angle between the positive x-axis and the Sun vector projection. The slew angle estimate can be calculated from the Sun vector estimate using Equation 5.22:

$$\hat{\theta}_s = \tan^{-1} \left(\frac{\hat{s}_y}{\hat{s}_x} \right). \quad (5.22)$$

As shown in Equation 5.22, the slew angle estimate is calculated as a function of the arctangent of the x and y Sun vector components \hat{s}_x and \hat{s}_y , respectively. In practice, the arctangent function is corrected to account for the slew angle quadrant and converted to a counter-clockwise displacement from the positive x-axis in degrees.

In the following subsection, the criterion used to select an artificial light source for the RSS experimental platform are discussed as the next step.

5.4.2 Solar Simulator Literature Review

In the literature, the most commonly considered characteristics to simulate sunlight conditions are spectral matching, intensity matching, temporal stability, spatial uniformity, and field-of-view (FOV) [92,93]. Spectral matching and intensity matching refer to a light source's ability to reproduce the power spectral distribution and total incident intensity of a reference solar spectrum, respectively. Temporal stability refers to the stability of incident light intensity over time. Spatial uniformity refers to the uniformity of incident light intensity across a test plane area. Finally, the FOV of a light source is a measure of the maximum angle between light rays; for sunlight incident to Earth, the FOV is approximately 0.53 degrees [62].

Solar simulators may be used for a variety of both terrestrial and extraterrestrial applications [93,94]. Terrestrial applications aim to simulate sunlight as observed from Earth's surface; examples include the testing of photovoltaic systems as in [94,95]. Extraterrestrial applications, such as the Sun sensor testing completed in this research, aim to simulate sunlight as observed from space. Similar methodology and criterion may be used for both terrestrial and extraterrestrial applications; however, the characteristics of sunlight will be different as a result of the spectral absorption of Earth's atmosphere [96].

The air mass zero (AM0) spectrum represents the standard for simulating the conditions of sunlight in space [97].

The spectral power distribution for the AM0 solar spectrum is shown in Figure 28; this figure represents the distribution of light intensity across the AM0 spectral range, which includes contributions from the ultraviolet and infrared radiation bands in addition to visible light. Total solar irradiance is a measure of the total incident light intensity from Sun over the entire spectral range; for the AM0 spectrum, the total solar irradiance is equal to 1361.1 W/m^2 , which represents the area under the curve in Figure 28.

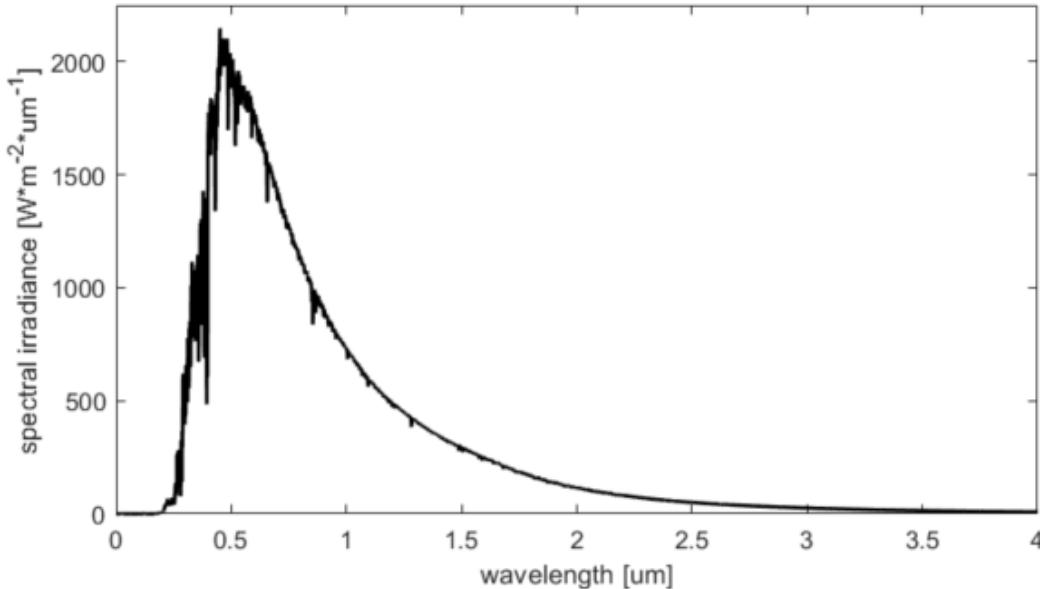


Figure 28. AM0 solar power spectral distribution (NOTE: this plot was generated using data from [97]).

Solar irradiance in the units of W/m^2 is a radiometric quantity by definition; however, it is often the case that solar intensity is instead specified as light illuminance in the units of lux. Illuminance, which is a photometric quantity, is a measure of light intensity as perceived by the human eye [93].

There is no simple conversion between solar irradiance and light illuminance. In a study done by Michael et al., 2020, it was shown that both online resources and peer reviewed journals show significant disagreement between conversion factors [98]. With the goal of producing a consistent rule-of-thumb conversion factor, they completed three separate trials utilizing published data and mathematical analysis, solar simulator measurements, and outdoor sunlight measurements. Their study yielded a final conversion factor of 120 lux per 1 W/m^2 for the AM1.5 solar spectrum. Assuming this conversion

factor is similar in magnitude for the AM0 spectrum, the solar constant of 1361.1 W/m^2 equates to an illuminance of approximately 160,000 lux.

Select applications, such as the photovoltaic cell testing discussed in [95], prioritize intensity and spectral requirements in their selection of a light source. For Sun sensor testing applications however, it is often the case that less stringent requirements are used [56,62]. In-depth reviews of different light sources and optical setups used for solar simulators are readily available in the literature [93,94,95] and will therefore not be repeated here. Rather, a brief comparison of existing setups used specifically for testing Sun sensors will be given.

In [56], analog Sun sensors are calibrated, and their performance characterized, using a set of red LEDs at intensity levels of 47000 lux and 88500 lux. The authors provide an in-depth description of the necessary steps used to convert their measurements from photometric intensity units of lux to standard radiometric intensity units of W/m^2 ; furthermore, their results are validated using both a secondary artificial light source and early results from the UPMSat-2 mission.

In [62], a fully integrated ADCS testing platform is described. The authors specify the spectral and intensity characteristics of their setup in terms of photometric units. Their platform uses a COTS LED studio light with a Fresnel lens attachment for light collimation and is capable of producing white light to an intensity of 130000 lux with a FOV which is less than 0.53 degrees (the Sun's angular diameter as observed from Earth). A testing campaign was completed to validate the FOV, spatial uniformity, and temporal stability of their solar simulator setup. The authors make a note of the limitations of using white light as it cannot fully reproduce the UV and IR bands of the AM0 solar spectrum; however, this was not considered a limiting factor in their application as the response of sensors used on their platform are mostly sensitive to the visible bands of light.

For this research, the goal was to select a relatively low-cost COTS light source capable of approximating AM0 sunlight for an initial validation of the RSS experimental platform for ADS testing. As discussed in the following subsection, the photodiodes used on LORIS are ambient light sensors which are primarily sensitive to the visible light spectrum. For this reason, higher priority was given to the light source's photometric light intensity, rather than its ability to reproduce the entire AM0 solar spectrum. Additionally,

the selection of a light source was limited to LED technology due to its COTS availability, low-cost, and high temporal stability [94,95]. To minimize conflicting photodiode measurements, it was also a goal to select a light source with minimal FOV; however, because collimating lens may be easily added for future research, this was not given high priority for the goals outlined in this thesis.

5.4.3 Sun Sensor Test Apparatus

The original design for the LORIS Sun sensor is discussed in [1] where a pyramidal arrangement of three photodiodes mounted on a custom mask was used to optimize the sensor FOV. This Sun sensor mask has since gone through several minor design changes which are reflected in the 3D printed version used in this research. The 3D printed Sun sensor assembly is shown pictured in Figure 29.

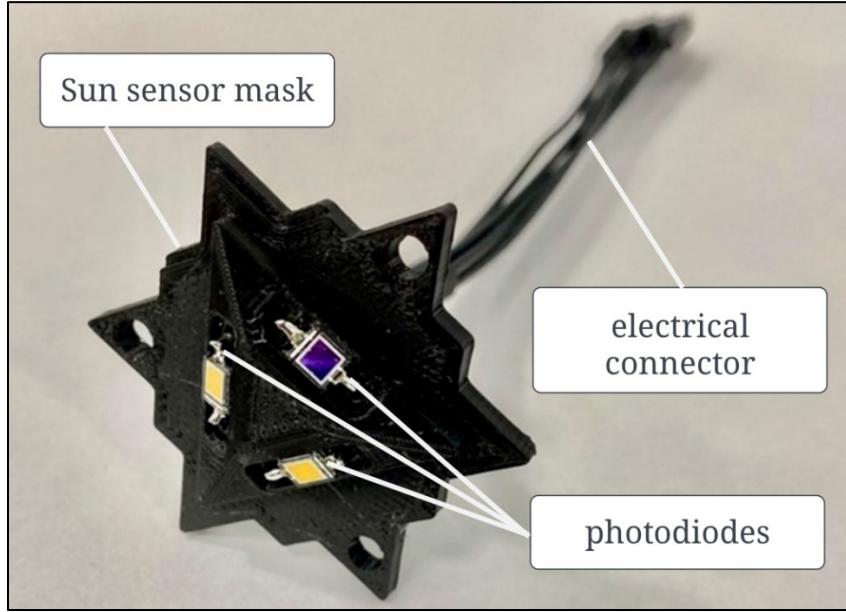


Figure 29. 3D Printed version of the LORIS Sun sensor used on the RSS platform (NOTE: the pictured photodiodes only appear to be different colors as a result of the camera angle and reflected light).

In Figure 29, one of the 3D printed Sun sensor masks used on the RSS is shown with three attached photodiodes and its electrical connector. Similar to the Sun sensor in [1], photodiodes are evenly spaced with 120-degree separation along the mask normal and 60-degree inclination from the horizontal plane. Four Sun sensor masks are used on the RSS platform, with one placed on each of the RSS faces which are perpendicular to the axis of rotation.

The COTS photodiodes used in this research were previously selected for the LORIS satellite and are the SFH-2430 ambient light sensors from Osram [99]. A custom transimpedance amplifier circuit designed for the LORIS satellite was used on the RSS platform to transform current intensities into voltages which are measured by the RSS' payload microcontroller.

The selected light source to simulate LEO sunlight conditions in this research is the COTS Forza 200 LED Monolight from Nanlite. With an added Fresnel lens attachment, the Forza 200 can produce white light to a maximum intensity of 126413 lux at 1 meter [100]. The Fresnel lens has an adjustable FOV with a minimum beam angle of 10 degrees; this FOV, which is greater than the 0.53 beam angle of sunlight in LEO, is considered one of the limiting factors of this test setup. As discussed in Section 5.4.2, adding a secondary collimating lens will reduce the light source FOV; however, due to project timeline and budgetary constraints, this was outside of the scope of the Sun sensor tests completed in this research.

A similar test setup to the one used for the RSS slew controller tests in Figure 23 was used to test the Sun sensors here. A picture of the modified setup is shown in Figure 30.

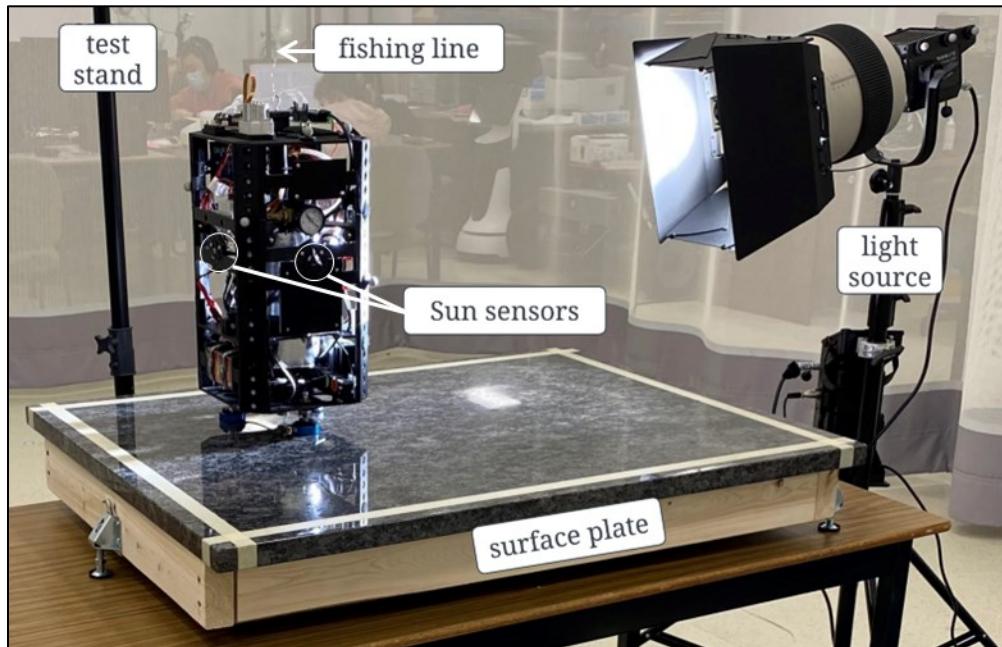


Figure 30. RSS Sun sensor experimental testbed, including RSS with fishing line attachment, granite platform, and Forza 200 light source with Fresnel lens and barn door attachments.

In Figure 30, the RSS platform is pictured using the previously described slew controller setup with the addition of the Forza 200 light source. Prior to the start of the experimental tests, the light source was pointed towards the center of the x-plus Sun sensor mask with an arbitrary elevation angle α_s as shown in Figure 31 a) and b). Light intensity measurements were recorded while the RSS was still stationary to align the 2D Sun vector slew angle estimate with the MOCAP ground truth attitude. The position of the light source was held constant for the duration of the experiment and the slew angle estimate was calculated as the counter-clockwise angular displacement between the RSS BF positive x-axis and the light source light vector as shown in Figure 31 c).

In the following subsection, the Sun vector determination experiments completed with the RSS test setup are discussed.

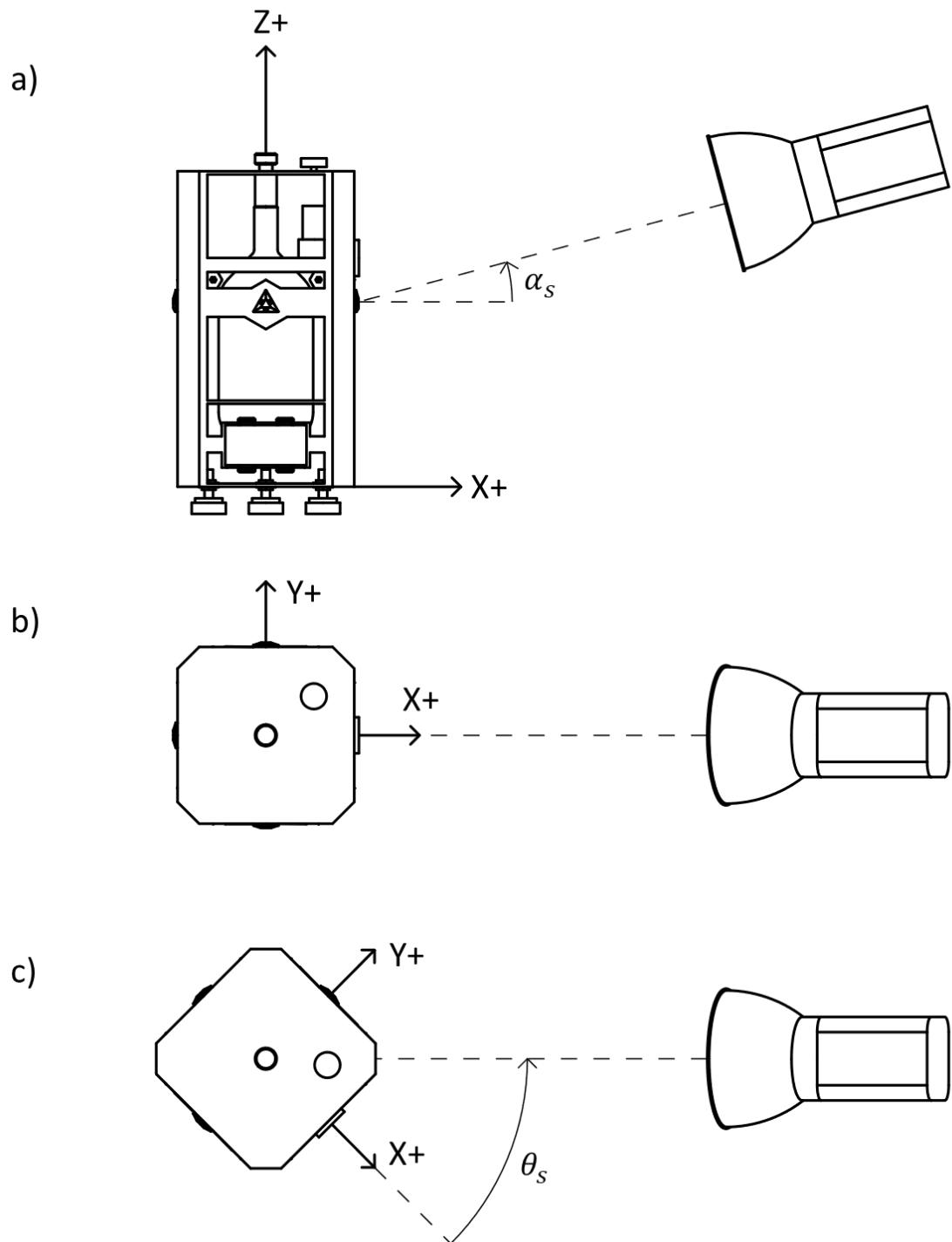


Figure 31. RSS Sun sensor testing configuration: a) side view of test setup illustrating light source elevation angle α_s w.r.t. RSS Z-X plane, b) top view of test setup illustrating the initial position of the light source for aligning the Sun vector estimate with ground truth attitude, c) top view of test setup illustrating the RSS slew angle θ_s measurement convention.

5.4.4 RSS Sun Vector Determination Experiment

For the Sun vector determination experiment, the RSS was manually actuated to an arbitrary slew rate. Photodiode light intensity and MOCAP ground truth measurements were time stamped and recorded by the OBC while the RSS completed several full rotations over an approximate two-minute period. The ground truth attitude information was used to estimate the slew rate of the RSS over time. The MOCAP ground truth slew angle and estimated slew rate are plotted against time in Figure 32 a) and b), respectively.

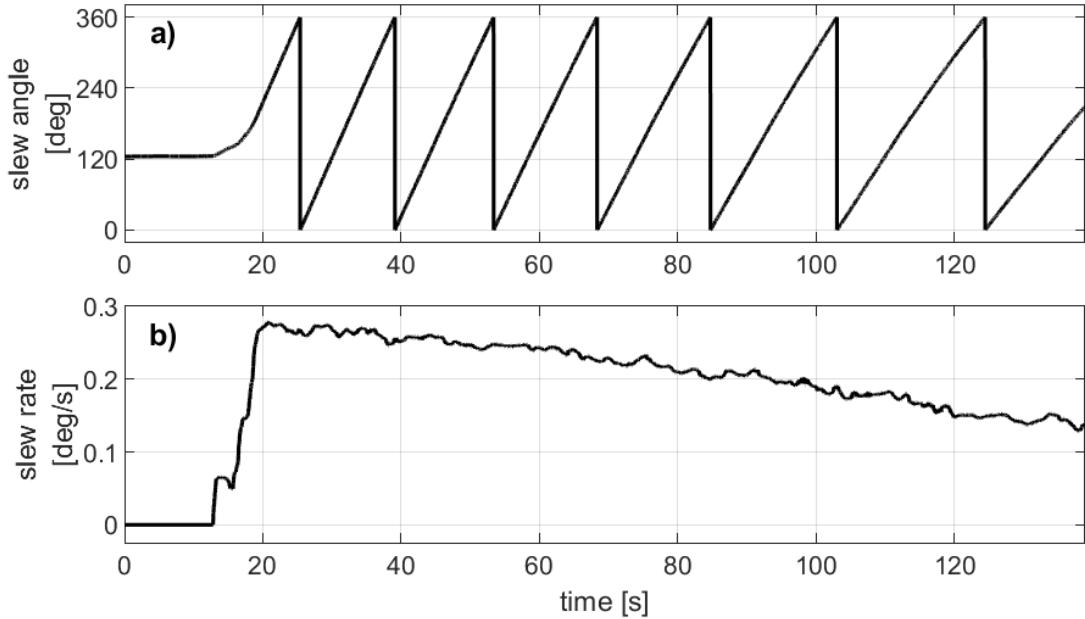


Figure 32. RSS Sun sensor experiment slew trajectory: a) RSS ground truth slew angle from MOCAP measurements aligned with Sun vector estimate and b) estimated RSS slew rate calculated from ground truth attitude vs. time. The RSS slowly decelerates over the course of seven full 360-degree rotations.

Outliers were removed from the raw photodiode intensity measurements using MATLAB's `isoutlier()` function. It detected outliers which were more than three median absolute deviations from the moving local median for every vector of 10 data points [80]. The 2-norm was calculated for each set of three photodiodes to reflect the combined intensity for the respective Sun sensor masks. As each mask corresponds to a different RSS BF direction, combined mask intensities were easily correlated to the RSS slew angle at any given time. Both the raw and filtered 2-norm intensity for each mask are plotted against time in Figure 33 where filtered signals are color coded according to the RSS BF direction they correspond to, and raw 2-norm signals are plotted as dashed black lines. Furthermore,

the signals in Figure 33 are all normalized relative to the maximum recorded filtered signal magnitude.

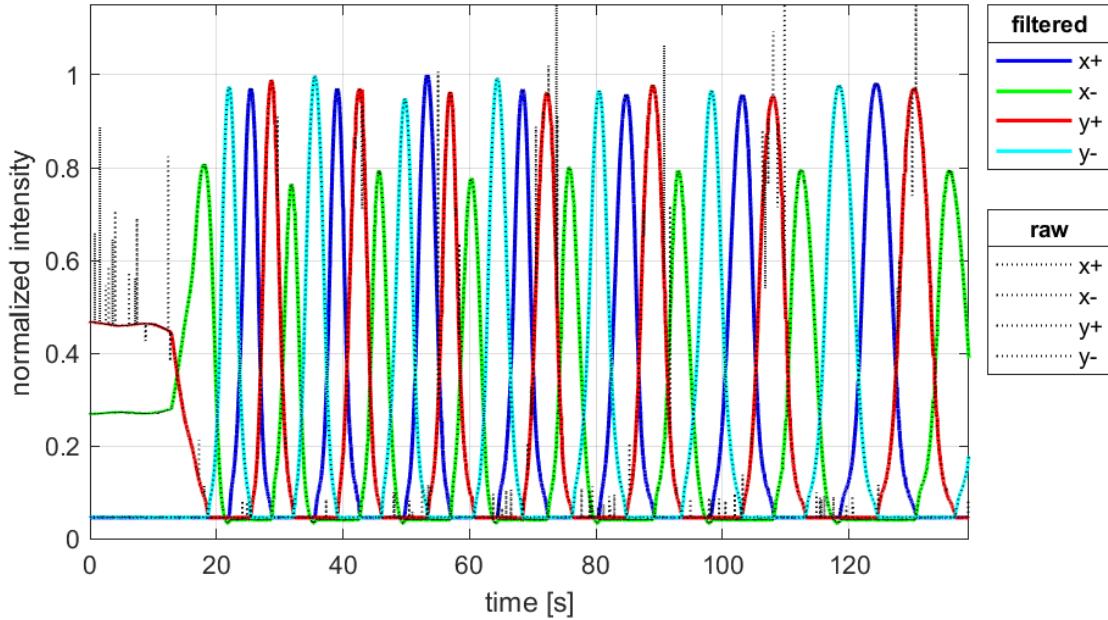


Figure 33. Sun sensor mask intensities: raw vs. filtered (NOTE: signals represent the normalized mask intensities for each set of three photodiodes which correspond to an RSS BF axis direction).

Visual inspection of the filtered signals in Figure 33 shows that outliers were removed from the raw intensity measurements. There are peaks for each of the combined intensity signals which occur when the light source is perpendicular to each mask. In a theoretical test environment with uniform light source intensity and equally sensitive photodiodes, the intensity peaks shown in Figure 33 should all be of the same magnitude; however, this was not observed in practice.

The x-minus mask intensity peaks are significantly lower than the others; this was found to be a result of a faulty photodiode on the x-minus mask. Other discrepancies between mask intensities may be attributed in part to differing photodiode sensitivities; however, intensity peaks for individual masks were also inconsistent with time. The reason for temporal inconsistencies may be due to the combined effects of small translational drift that occurred during the experiment and the non-uniform light intensity which was in turn a result of the 10-degree light source beam angle.

All of the photodiodes used in this research were assumed to have equal light sensitivity. The filtered intensity measurements were normalized by the maximum photodiode intensity observed over the entire experiment. A minimum intensity threshold was used to reject intensity measurements which resulted primarily from external sources such as ambient light conditions, reflections off the granite surface plate, or electrical bias from photodiode dark currents. A sensitivity analysis was conducted to identify the minimum intensity threshold which produced the most accurate and consistent results.

For each of the minimum intensity thresholds tested, the methodology described in Section 5.4.1 was used to estimate Sun vectors from the filtered experimental photodiode intensity measurements. At each time step, photodiodes were categorized according to their measured intensities, where photodiodes with intensities above, or below, the set threshold are referred to as active or inactive, respectively. Only active photodiode intensity measurements were used to estimate Sun vectors using Equation 5.21. Furthermore, Sun vector estimates were only generated for time instances where there were at least three active photodiodes. Time instances in which no Sun vector estimates were generated were tracked and the dropout frequency was calculated as a percentage of the total experiment duration. The accuracy of 3D Sun vector estimates were approximated based on the error between the 2D Sun vector projection used to estimate the RSS slew angle and the MOCAP ground truth measurements. The final RMSE and dropout frequency of the estimated slew angles were the primary metrics for evaluating different minimum intensity thresholds.

Minimum intensity thresholds were specified as a percentage of the maximum intensity reading used to normalize the filtered photodiode measurements. Detailed results from the sensitivity analysis are provided in Table 18 of Appendix C.3.3.

In Figure 34, trend plots for the slew angle final RMSE and dropout frequency are shown for minimum intensity thresholds between 0 – 30%. For thresholds under 3.8%, the error statistics show little to no change and estimation dropouts never occur. A local minimum was observed for the 4% threshold test which resulted in a final RMSE of 17.5 degrees and no estimation dropouts. Tests with thresholds between 4 – 26% exhibit fairly constant trends with steadily increasing RMSE and dropout frequency observed. Finally, the slew estimate diverges for thresholds above 27%.

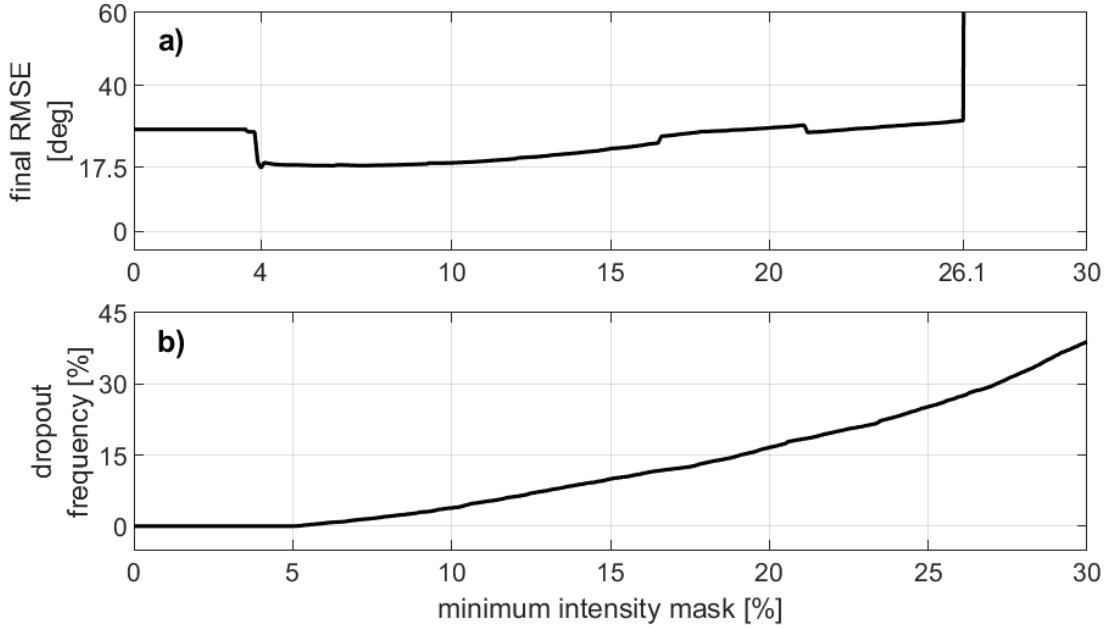


Figure 34. Sun sensor intensity threshold sensitivity analysis trends: a) final slew angle estimate RMSE and b) Sun vector estimate dropout frequency vs. intensity threshold. A local minimum was observed for the 4% intensity threshold test which resulted in the lowest final RMSE of 17.5 degrees and no estimation dropouts.

The 4% threshold test showed the best performance when considering the previously mentioned performance metrics. However, closer inspection of the results indicated some high frequency oscillations of photodiode states.

In Figure 35, the active state for a single photodiode is plotted against time for the first 45 seconds of the experiment for 4% and 4.5% minimum intensity threshold tests. Increasing the threshold from 4% to 4.5% eliminated the high frequency photodiode state oscillations. The oscillations in Figure 35 likely indicate that the 4% threshold is approximately equal to the combined intensity of the ambient light conditions and photodiode dark current for the experimental measurements. To avoid the use of erroneous photodiode intensity measurements, a 4.5% minimum intensity threshold was selected for further analysis of the collected experimental data.

The resulting slew estimate, ground truth attitude, and slew estimation error for the 4.5% threshold test are shown in Figure 36.

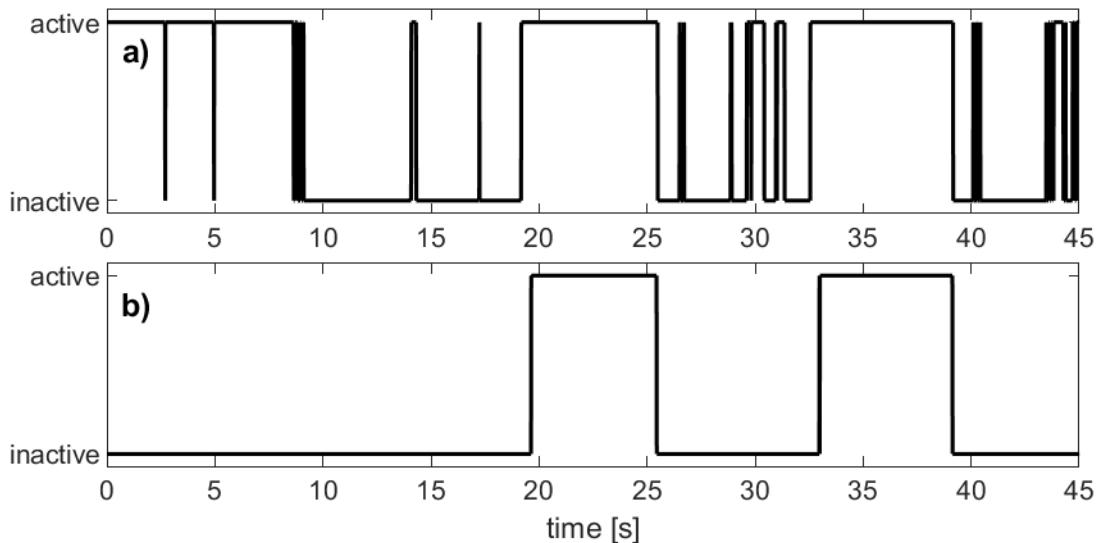


Figure 35. Sun sensor state comparison: single photodiode state vs. time from Sun vector determination experimental data using a) 4.0% minimum intensity threshold and b) 4.5% minimum intensity threshold vs. time. Increasing the intensity threshold slightly above the 4% local minimum observed in Figure 34 eliminates high frequency photodiode state oscillations and erroneous intensity measurements.

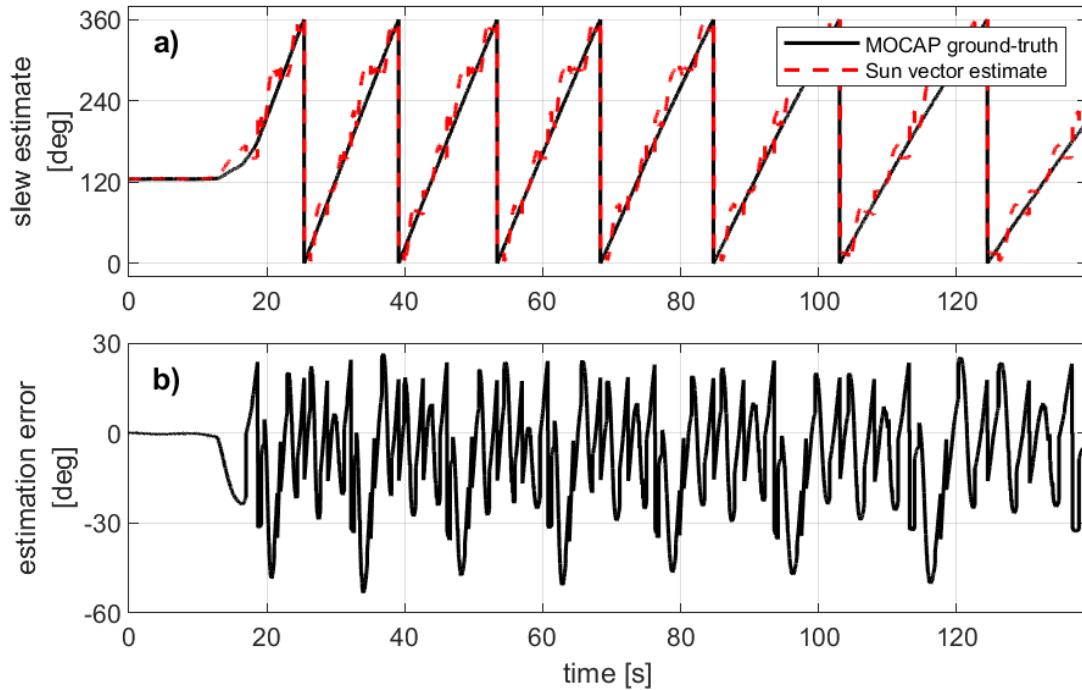


Figure 36. Sun sensor slew angle estimation summary: a) estimated and ground truth slew angles and b) Sun sensor slew angle estimation error vs. time. Note that the cyclic errors observed in the slew angle estimates are likely correlated to the RSS' ground truth attitude.

The results show that the Sun sensor slew angle estimate exhibits cyclic errors compared to the MOCAP ground truth. The absolute maximum slew angle estimation error observed was 53.34 degrees. Furthermore, the slew angle estimation error final RMSE and standard deviation are equal to 18.34 degrees and 16.91 degrees, respectively. The resulting slew angle estimation errors, which are a 2D approximations of the 3D Sun vector estimation errors, are larger than the expected accuracy of typical photodiode-based Sun sensors (approximately 1 – 3 degrees [91]). Contributions to these large errors are likely due, in part, to the test setup used.

To further illustrate the correlation between RSS slew angle and photodiode sensor measurements, select results were tracked as a function of the ground truth attitude over the entire experiment. The average for several full RSS rotations was then taken for each of the tracked variables. The final results are illustrated as polar plots in Figure 37. Plot a) shows the average attitude error for each slew angle degree of rotation. Plot b) shows the average total intensity for each slew angle degree of rotation. The average total intensity is equal to the sum of the normalized intensity readings from active photodiodes, represented as a percentage. Plot c) shows the average number of active photodiodes for each slew angle degree of rotation.

All three plots in Figure 37 show approximately symmetrical results between quadrant I (0 – 90 deg) and quadrant IV (270 – 360 deg), indicating similar performance from the x-plus, y-plus, and y-minus Sun sensor masks. Error minima are generally observed when the light source is oriented 0, 15, and 45 degrees relative to the Sun sensor masks. Not surprisingly, intensity peaks occur when the light source is perpendicular to the Sun sensor masks; however, the best viewing geometry occurs when the light source is in-between two masks, which generally results in a maximum of 6 active photodiodes. As previously discussed, the x-minus Sun sensor mask has one non-functioning photodiode. This photodiode is angled towards the y-minus RSS BF axis. The impacts of this faulty photodiode are reflected in all three plots from Figure 37. In plot a) there is a large error spike near the 230-degree marker which can be correlated to the reduced average total intensity around this region in plot b).

Replacing the faulty photodiode on the x-minus Sun sensor mask will likely result in symmetrical results about the x- and y- axes. However, even with fully functioning

photodiodes, error spikes up to 26.83 degrees still occur in quadrant I and quadrant IV. Adding a Sun sensor mask to the RSS BF z-plus face should lead to improved accuracy as a result of an increased number of active photodiodes for each Sun sensor estimate.

As previously discussed, the main deficiencies of the current test setup are related to spatial and temporal inconsistencies which result from the combined effects of small translational drift of the RSS platform and the light source FOV. Building a simulation model of the test setup to characterize the impacts of these inconsistencies is a necessary task to validate the use of the RSS experimental platform for performance characterization of Sun sensors; however, the current setup may still be used for observing overall performance trends. The results shown here demonstrate the utility of the RSS platform for ADS testing; however, additional improvements to the setup are necessary to accurately quantify Sun vector estimation accuracy.

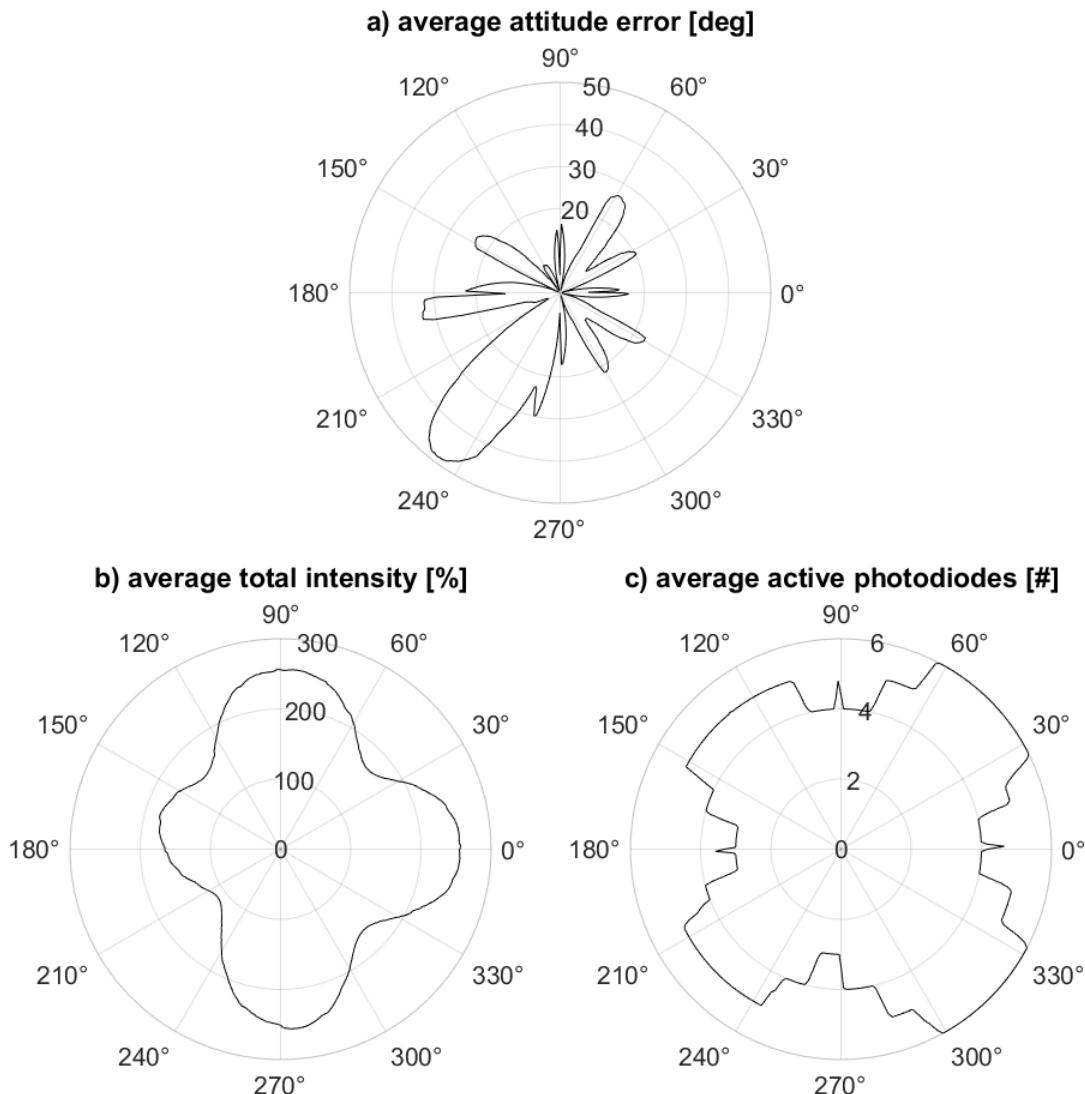


Figure 37. Sun sensor performance polar plots: ground truth attitude vs. a) average Sun sensor attitude estimation error, b) average total photodiode measurement intensity, c) average number of active photodiodes (NOTE: 0-degree and 90-degree angles correspond to when the light source is directed toward the RSS BF x-plus axis direction and y-plus axis direction, respectively). There is a loss of symmetry between the four quadrants due to the faulty photodiode on the x-minus mask; this also results in the large error spike seen in plot a) at an angle of approximately 230 degrees.

Chapter 6. Conclusions and Recommendations

This thesis presents newly developed tools to design, test, and validate GNC strategies for CubeSat Earth observation missions. Motivation for this research was in part to continue work previously completed in [1] and to advance the capabilities of the DSSL for CubeSat missions. The scope of this research was limited to two components of the ADCS simulator which required additional realism: 1. the orbit propagator and 2. the ADCS sensor and actuator models.

The ODS simulator and RSS testbed are effective tools which can be used in combination with the ADCS simulator [1] to develop new GNC strategies for CubeSat missions. The results of the ODS simulator and ADCS device testing completed in this research are discussed in Section 6.1 and Section 6.2, respectively. Then, a summary of the contributions from this thesis and recommendations for future work are provided in Section 6.3.

6.1 ODS Simulator Testing Results

The ODS simulator was developed as a tool to design and test onboard ODS strategies. The goals for the ODS simulator development were to 1. reproduce real-time GPS measurements from a satellite mission in LEO, 2. implement and test navigation algorithms commonly used for onboard orbit determination, and 3. make recommendations for an ODS strategy which is suitable for CubeSat Earth observation missions with limited onboard resources.

Three commonly used navigation algorithms were implemented into the ODS simulator and tested using real measurements from the GRACE-FO satellite mission. The SGP4 algorithm provided coarse estimation accuracy as expected with a 24-hour position accuracy of 821.4 m; however, the algorithm does not rely on GPS and can maintain moderate accuracy with infrequent TLE updates. The GPS SPS produced the best 24-hour position accuracy of 21.54 m; however, solutions are only generated as GPS updates were available and were highly sensitive to the quality of the measurements.

The RDEKF algorithm was found to be the most robust option for POD; however, its performance is dependent on its setup configuration. The fastest convergence was found when position and velocity initial conditions were supplied from the GPS SPS and SGP4

algorithms, respectively. The TB and TBJ2 algorithm configurations produced similar results for 100% GPS duty cycles; however, the TBJ2 model was more robust to infrequent GPS updates with a 24-hour position accuracy of 73.72 m for a GPS duty cycle of 22% (i.e. GPS turned on for 3 minutes once every 13.5 minutes).

To conserve onboard power and maintain position and velocity estimation accuracy, the OGNS algorithm was developed to switch between coarse and fine ODS algorithms. Custom target prediction algorithms were developed based on the coarse search phase approach from [82]. These algorithms were used to automatically switch between the SGP4 algorithm and TB RDEKF algorithm configuration during target flybys. The OGNS algorithm was tested for three different target locations over the same 4-day period. Only 2 target flybys were missed out of 42, one of which was due to insufficient startup information. The average position accuracy for target flybys was less than 30 m and the GPS duty cycle less than 1%. These results show the OGNS algorithm efficiently switches between coarse and fine ODS strategies, resulting in a significant decrease of onboard power consumption from the GPS receiver.

The OGNS algorithm can, and should, be configured differently to a specific satellite mission. Various combinations of ODS strategies may be used to achieve different accuracy requirements. Performance of ODS navigation algorithms, as well as the target prediction algorithms, may depend on the GPS receiver used, orbit shape, and more. To validate a specific OGNS setup configuration for any mission type, additional testing should be conducted to ensure the performance meets mission requirements. Further development of the OGNS algorithm may also be considered to improve onboard autonomy for other satellite subsystems.

6.2 ADCS Device Testing Results

The RSS testbed was used to complete experiments with a custom RWA and Sun sensor which were designed for the LORIS satellite mission. The primary objectives were to identify accurate models for the ADCS simulator, characterize the performance of the custom devices for future satellite missions, and evaluate the capabilities of the RSS testbed to test ADCS components and control strategies.

MATLAB's *fminsearch()* algorithm was used with experimental step input responses to identify parametric motor models for the RWA based on the viscous friction and viscous-Coulomb friction motor model definitions. Models which were identified based on viscous-Coulomb friction parameters did a much better job simulating outputs across the entire RWA nominal operating range. The best performing model resulted in an average fit of 88.4%.

The RWA was unable to meet its original momentum storage design requirement for the LORIS satellite with a maximum speed range of -1800 rpm to 1800 rpm, maximum momentum storage of 1.906E-3 N m s starting from rest, and a maximum continuous acceleration torque of 1.157E-3 N m. It is recommended that future designs of the RWA either address the significant friction or use a larger motor to increase momentum storage.

The identified RWA motor model was used in combination with the RSS attitude control simulator to design a slew controller which was tested using the RSS experimental testbed for 20-degree and 30-degree slew tests. The results showed excellent agreement between simulation predicted and experimental measurements validating the identified RWA motor model.

Furthermore, a Sun vector determination experiment was conducted using the RSS experimental testbed, custom LORIS Sun sensor photodiode arrangements, and a COTS solar simulator. Overall performance trends were observed for the LORIS Sun sensors; however, error contributions which resulted from a faulty photodiode, RSS translational drift, and the light source 10-degree FOV need to be addressed for the RSS experimental testbed to be used for absolute performance characterization of the Sun sensors.

6.3 Contributions

Contributions from this research are:

The first contribution of this research is the OGNS algorithm which can be used on onboard ODS to ensure fine orbital accuracy (< 200 m) during target flybys while minimizing onboard GPS usage. The OGNS combines traditional ODS strategies and an analytical target prediction algorithm to automatically switch between coarse and fine orbit determination modes during target flybys. A custom ODS simulator was developed to test the OGNS algorithm with ground truth ephemeris and raw pseudorange measurements

from the GRACE-FO mission and it successfully reduced GPS duty cycle to less than 1% for three separate target locations. **The OGNS algorithm is a computationally efficient alternative to traditional methods which use sophisticated force models to reduce onboard GPS usage.** In this research, the SGP4 and RDEKF algorithms were used; however, different implementations of the OGNS algorithm may also be used to meet specific mission requirements.

The second contribution of this research is the RSS experimental testbed for ADCS testing. The RSS platform uses planar air bearings as opposed to typical setups which use spherical air bearing platforms. System identification experiments were conducted to identify a motor model for the custom LORIS RWA which were then validated using the RSS testbed. Then, tests completed with the RSS testbed illustrated overall trends for custom LORIS Sun sensors; improvements to the setup will allow for absolute performance characterization. **These tests show the value of planar air bearing platforms to test and validate ADCS equipment. Such testbeds that can both translate and be used for ADCS testing are relatively inexpensive to mitigate CubeSat development risk compared to the cost of mistakes discovered in-orbit.**

6.4 Future Work

Recommendations for future research include:

1. continued development of the ODS simulator, such as:
 - a. improving the GPS sensor model to include simulated startup delays
 - b. including ground truth experimental data from other satellite missions
 - c. adding features to the OGNS algorithm for improved satellite autonomy
2. continued development of the RSS testbed, such as:
 - a. characterizing the impacts of the RSS drift and Sun lamp FOV on Sun sensor readings
 - b. development of magnetometer and magnetorquer testing strategies
3. continued development of the ADCS simulator based on results from the ODS simulator tests and ADCS experiments

References

- [1] Wailand, A. (2020). *Development of a Computer Simulation Tool to Study the Attitude Determination and Control of CubeSats*. Dalhousie University. <http://hdl.handle.net/10222/80095>
- [2] Wilson, W. R., Jones, L. L., & Peck, M. A. (2013). *A Multimodule Planar Air Bearing Testbed for CubeSat-Scale Spacecraft*. Journal of Dynamic Systems, Measurement, and Control, 135(4). <https://doi.org/10.1115/1.4023767>
- [3] Sabatini, M., Palmerini, G. B., & Gasbarri, P. (2015). *A testbed for visual based navigation and control during space rendezvous operations*. Acta Astronautica, 117, 184–196. <https://doi.org/10.1016/j.actaastro.2015.07.026>
- [4] Zappulla, R., Virgili-Llop, J., Zagaris, C., Park, H., & Romano, M. (2017). *Dynamic Air-Bearing Hardware-in-the-Loop Testbed to Experimentally Evaluate Autonomous Spacecraft Proximity Maneuvers*. Journal of Spacecraft and Rockets 2017 54:4, 825-839. <https://doi.org/10.2514/1.A33769>
- [5] Hovell, K. & Ulrich, S. (2018). *Postcapture Dynamics and Experimental Validation of Subtethered Space Debris*. Journal of Guidance, Control, and Dynamics 2018 41:2, 519-525. <https://doi.org/10.2514/1.G003049>
- [6] Curtis, H. D. (2014). *Orbital Mechanics for Engineering Students* (3rd Edition). Butterworth-Heinemann. <https://doi.org/10.1016/C2011-0-69685-1>
- [7] Parkinson, B. W. (1996). *Global Positioning System: Theory and applications* (Vol. 1). American Inst. of Aeronautics and Astronautics. <https://doi.org/10.2514/4.866388>
- [8] Blewitt, G. (2000). *Basics of the GPS Technique: Observation Equations*. <https://nbgmg.unr.edu/staff/pdfs/blewitt%20basics%20of%20gps.pdf>
- [9] Weston, S., Benavides, G., Krage, F., Hines, J., Mauro, S., Etchey, S., Kiara O'Neill, & Braun, B. (2021). *State-of-the-Art Small Spacecraft Technology*. NASA STI Repository (NTRS).
- [10] Poghosyan, & Golkar, A. (2017). *CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions*. Progress in Aerospace Sciences, 88, 59–83. <https://doi.org/10.1016/j.paerosci.2016.11.002>
- [11] Villela, Costa, C. A., Brandão, A. M., Bueno, F. T., & Leonardi, R. (2019). *Towards the Thousandth CubeSat: A Statistical Overview*. International Journal of Aerospace Engineering, 2019, 1–13. <https://doi.org/10.1155/2019/5063145>
- [12] Langer, Weisgerber, M., Bouwmeester, J., & Hoehn, A. (2017). *A reliability estimation tool for reducing infant mortality in Cubesat missions*. 2017 IEEE Aerospace Conference, 1–9. <https://doi.org/10.1109/AERO.2017.7943598>

- [13] Alanazi, & Straub, J. (2019). *Engineering Methodology for Student-Driven CubeSats*. Aerospace, 6(5), 54–. <https://doi.org/10.3390/aerospace6050054>
- [14] Monteiro, Rocha, R. M., Silva, A., Afonso, R., & Ramos, N. (2019). *Integration and Verification Approach of ISTSat-1 CubeSat*. Aerospace, 6(12), 131–. <https://doi.org/10.3390/aerospace6120131>
- [15] Montenbruck, O., Garcia-Fernandez, M., & Williams, J. (2006). *Performance comparison of semicodeless GPS receivers for LEO satellites*. GPS Solutions, 10(4), 249–261. <https://doi.org/10.1007/s10291-006-0025-9>
- [16] Hauschild, A., Markgraf, M., & Montenbruck, O. (2014). *The Navigation and Occultation eXperiment – GPS Receiver Performance On Board a LEO Satellite*. Inside GNSS, pp. 48–57.
- [17] Fang, B. & Seifert, E. (1985). *An evaluation of Global Positioning System data for Landsat-4 orbit determination*. 23rd Aerospace Sciences Meeting. <https://doi.org/10.2514/6.1985-286>
- [18] Fu, L. L., Christensen, E. J., Yamarone, C. A., Lefebvre, M., Menard, Y., Dorrer, M., & Escudier, P. (1994). *TOPEX/POSEIDON mission overview*. Journal of Geophysical Research, 99(C12), 24369–24381. <https://doi.org/10.1029/94JC01761>
- [19] Ijssel, J. V. D., Visser, P., & Patiño Rodriguez, E. (2003). *Champ precise orbit determination using GPS data*. Advances in Space Research, 31(8), 1889–1895. [https://doi.org/10.1016/S0273-1177\(03\)00161-3](https://doi.org/10.1016/S0273-1177(03)00161-3)
- [20] Choi, K. R., Ries, J. C., & Tapley, B. D. (2004). *Jason-1 Precision Orbit Determination by Combining SLR and DORIS with GPS Tracking Data*. Marine Geodesy, 27(1-2), 319–331. <https://doi.org/10.1080/01490410490465652>
- [21] Lee, B. S., Yoon, J., Hwang, Y., & Kim, J. (2005). *Orbit determination system for the KOMPSAT-2 using GPS measurement data*. Acta Astronautica, 57(9), 747–753. <https://doi.org/10.1016/j.actaastro.2005.03.066>
- [22] Kang, Z., Tapley, B., Bettadpur, S., Ries, J., Nagel, P., & Pastor, R. (2006). *Precise orbit determination for the GRACE mission using only GPS data*. Journal of Geodesy, 80(6), 322–331. <https://doi.org/10.1007/s00190-006-0073-5>
- [23] Kovar, P. (2017). *piNAV L1—GPS receiver for small satellites*. Gyroscopy and Navigation, 8(2), 159–164. <https://doi.org/10.1134/S2075108717020079>
- [24] Grelier, T., Ries, L., Bataille, P., Perrot, C., & Richard, G. (2012). *A new operational low cost GNSS Software receiver for Microsatellites*. 2012 6th ESA Workshop on Satellite Navigation Technologies (Navitec 2012) & European Workshop on GNSS Signals and Signal Processing, 1–5. <https://doi.org/10.1109/NAVITEC.2012.6423051>

- [25] Kovar, P., & Jelen, S. (2014). *Cold Start Strategy of the CubeSat GPS Receiver*. Advances in Electrical and Computer Engineering, 14(2), 29–34. <https://doi.org/10.4316/AECE.2014.02005>
- [26] Kahr, E., Bradbury, L., O’Keefe, K. P. G., & Skone, S. (2013). *Design and Operation of the GPS Receiver Onboard the CanX-2 Nanosatellite*. Navigation, 60(2), 143–156. <https://doi.org/10.1002/navi.36>
- [27] Rossouw, N. C. (2015). *A GPS-based On-board Orbit Propagator for Low Earth-Orbiting CubeSats*. Stellenbosch University.
- [28] Vetter, J. R. (2007). *Fifty Years of Orbit Determination: Development of Modern Astrodynamics Methods*. Johns Hopkins APL Technical Digest, 27(3), 239–252.
- [29] Selvan, K., Siemuri, A., Kuusniemi, H., & Valisuo, P. (2021). *A Review on Precise Orbit Determination of Various LEO Satellites*. CEUR Workshop Proceedings. <http://ceur-ws.org/Vol-2880/paper6.pdf>
- [30] Wildt, C. (2017). *Accuracy in Orbital Propagation: A Comparison of Predictive Software Models*. Naval Postgraduate School. <http://hdl.handle.net/10945/55555>
- [31] Montenbruck, O., & Gill, E. (2000). *Real-Time Estimation of SGP4 Orbital Elements from GPS Navigation Data*. International Symposium on Space Flight Dynamics.
- [32] Greene, M., & Zee, R. (2009). *Increasing the Accuracy of Orbital Position Information from NORAD SGP4 Using Intermittent GPS Readings*. 23rd Annual AIAA/USU Conference on Small Satellites.
- [33] Sha, L., Nikicio, A. N., Weiqi, T., Edwin, C., & Hiang, G. C. (2018). *GPS and SGP4 based Orbit Propagation Onboard the Galassia-2 Nanosatellite*. 4S Symposium. <https://scholarbank.nus.edu.sg/handle/10635/156577>
- [34] A. Mander. (2011). *Constrained GPS-Based Precise Orbit Determination of Low Earth Orbiters*. York University. http://sbisnath.apps01.yorku.ca/wp-content/uploads/2013/01/thesis_mander.pdf
- [35] Gill, E. & Montenbruck, O. (2004). *Comparison of GPS-Based Orbit Determination Strategies*. 18th International Symposium on Space Flight Dynamics.
- [36] Montenbruck O. (2003). *Kinematic GPS positioning of LEO satellites using ionosphere-free single frequency measurements*. Aerospace Science and Technology, 7(5), 396–405. [https://doi.org/10.1016/S1270-9638\(03\)00034-8](https://doi.org/10.1016/S1270-9638(03)00034-8)

- [37] Yunck, T. P., Wu, S. C., Wu, J. T., & Thornton, C. (1990). *Precise tracking of remote sensing satellites with the Global Positioning System*. IEEE Transactions on Geoscience and Remote Sensing, 28(1), 108–116. <https://doi.org/10.1109/36.45753>
- [38] Mander, A. & Bisnath, S. (2012). *GPS-based precise orbit determination of Low Earth Orbiters with limited resources*. GPS Solutions, 17(4), 587–594. <https://doi.org/10.1007/s10291-012-0303-7>
- [39] Montenbruck, O., Markgraf, M., Garcia-Fernandez, M., & Helm, A. (2007). *GPS for Microsatellites – Status and Perspectives*. In Small Satellites for Earth Observation (pp. 165–174). Springer Netherlands. https://doi.org/10.1007/978-1-4020-6943-7_15
- [40] Bock, H., Jäggi, A., Dach, R., Schaer, S., & Beutler, G. (2009). *GPS single-frequency orbit determination for low Earth orbiting satellites*. Advances in Space Research, 43(5), 783–791. <https://doi.org/10.1016/j.asr.2008.12.003>
- [41] Lantto, S. (2018). *Precise Orbit Determination of CubeSats Using Duty Cycled GPS Observations*. West Virginia University. <https://researchrepository.wvu.edu/etd/6035>
- [42] Wang, K., Allahvirdi-Zadeh, A., El-Mowafy, A., & Gross, J. N. (2020). *A Sensitivity Study of POD Using Dual-Frequency GPS for CubeSats Data Limitation and Resources*. Remote Sensing (Basel, Switzerland), 12(13), 2107–. <https://doi.org/10.3390/rs12132107>
- [43] Vigneron, A.C., de Ruiter, A. H. J., Burlton, B. V., & Soh, W. K. H. (2016). *Nonlinear filtering for autonomous navigation of spacecraft in highly elliptical orbit*. Acta Astronautica, 126, 138–149. <https://doi.org/10.1016/j.actaastro.2016.03.035>
- [44] Sun, X., Han, C., & Chen, P. (2017). *Real-time precise orbit determination of LEO satellites using single-frequency GPS receiver: Preliminary results of Chinese SJ-9A satellite*. Advances in Space Research, 60, 1478–1487.
- [45] Wesam, E. M., Zhang, X., Lu, Z., & Liao, W. (2017). *Kalman filter implementation for small satellites using constraint GPS data*. IOP Conference Series. Materials Science and Engineering, 211(1), 12015–. <https://doi.org/10.1088/1757-899X/211/1/012015>
- [46] Chiaradia, A. P. M., Kuga, H. K., & Prado, A. F. B. A. (2013). *Onboard and Real-Time Artificial Satellite Orbit Determination Using GPS*. Mathematical Problems in Engineering, 2013, 1–8. <https://doi.org/10.1155/2013/530516>
- [47] de Lafontaine, J., Buijs, J., Vuilleumier, P., van den Brambussche, P., & Mellab, K. (2000). *Development of the PROBA Attitude Control and Navigation Software*. 4th ESA International Conference on Spacecraft Guidance, Navigation and Control Systems.

- [48] Gill, E., Montenbruck, O., Brieß, K. (2000). *GPS-Based Autonomous Navigation for the BIRD Satellite*. International Symposium on Space Flight Dynamics.
- [49] Lee, B. S., Yoon, J., Hwang, Y., & Kim, J. (2005). *Orbit determination system for the KOMPSAT-2 using GPS measurement data*. Acta Astronautica, 57(9), 747–753. <https://doi.org/10.1016/j.actaastro.2005.03.066>
- [50] Montenbruck, O., Markgraf, M., Naudet, J., Santandrea, S., Gantois, K., & Vuilleumier, P. (2008). *Autonomous and Precise Navigation of the PROBA-2 Spacecraft*. AIAA/AAS Astrodynamics Specialist Conference and Exhibit. <https://doi.org/10.2514/6.2008-7086>
- [51] Bolandi, H., Larki, M. H. A., Abedi, M., & Esmailzade, M. (2013). *GPS based onboard orbit determination system providing fault management features for a LEO satellite*. Journal of Navigation, 66(4), 539–559. <https://doi.org/10.1017/S0373463313000179>
- [52] Brewer, M. (2012). *CubeSat Attitude Determination and Helmholtz Cage Design*. Air Force Institute of Technology. <https://scholar.afit.edu/etd/1030>
- [53] Vos, M. (2013). *Delfi-n3XT's Attitude Determination and Control Subsystem: Implementation and Verification of the Hardware and Software*. Delft University of Technology. <http://resolver.tudelft.nl/uuid:0640f4b3-8fc8-4d8f-9d8b-e47e6c98313d>
- [54] Farissi, M. S., Carletta, S., Nascetti, A., & Teofilatto, P. (2019). *Implementation and Hardware-In-The-Loop Simulation of a Magnetic Detumbling and Pointing Control Based on Three-Axis Magnetometer Data*. Aerospace, 6(12), 133–. <https://doi.org/10.3390/aerospace6120133>
- [55] Antonello, A., Olivier, L., & Francesconi, A. (2018). *Development of a low-cost Sun sensor for nanosatellites*. Acta Astronautica, 144, 429–436. <https://doi.org/10.1016/j.actaastro.2018.01.003>
- [56] Porras-Hermoso, A., Alfonso-Corcuera, D., Piqueras, J., Roibás-Millán, E., Cubas, J., Pérez-Álvarez, J., & Pindado, S. (2021). *Design, ground testing and on-orbit performance of a Sun sensor based on COTS photodiodes for the UPMSAT-2 satellite*. Sensors, 21(14), 4905. <https://doi.org/10.3390/s21144905>
- [57] Barreda Pupo, L. (2016). *Characterization of errors and noises in MEMS inertial sensors using Allan variance method*. Universitat Politècnica de Catalunya. <https://upcommons.upc.edu/handle/2117/103849?locale-attribute=en>
- [58] Zhang, Q., Wang, X., Wang, S., & Pei, C. (2018). *Application of Improved Fast Dynamic Allan Variance for the Characterization of MEMS Gyroscope on UAV*. Journal of Sensors, 2018, 1–6. <https://doi.org/10.1155/2018/2895187>

- [59] Hoevenaars, A. G. L. (2012). *Design, Integration and Verification of the Delft-n3Xt Reaction Wheel System*. Delft University of Technology.
<http://resolver.tudelft.nl/uuid:282ce049-a4e9-4e3d-84c6-8b16f9e5414d>
- [60] Carrara, V. (2010). *Experimental comparison between reaction wheel attitude controller strategies*. Journal of Aerospace Engineering, Sciences and Applications, 2(2), 1–9. <https://doi.org/10.7446/jaes.0202.01>
- [61] Carrara, V., & Kuga, H. K. (2013). *Estimating friction parameters in reaction wheels for Attitude Control*. Mathematical Problems in Engineering, 2013, 1–8. <https://doi.org/10.1155/2013/249674>
- [62] Modenini, D., Bahu, A., Curzi, G., & Togni, A. (2020). *A dynamic testbed for Nanosatellites Attitude Verification*. Aerospace, 7(3), 31. <https://doi.org/10.3390/aerospace7030031>
- [63] Gallina, A., Zagórski, P., & Rachucki, J. (2015). *Test Bed for the Simulation of Magnetic Field Measurements of Low Earth Orbit Satellites*. Mechanics and Control, 34(3), 47–. <https://doi.org/10.7494/mech.2015.34.3.47>
- [64] Chesi, S., Perez, O., & Romano, M. (2015). *A Dynamic, Hardware-in-the-Loop, Three-Axis Simulator of Spacecraft Attitude Maneuvering with Nanosatellite Dimensions*. JoSS, Vol. 4, No. 1, pp. 315–328.
- [65] Wolosik, A. T. (2018). *Advancements in the Design and Development of CubeSat Attitude Determination and Control Testing at the Virginia Tech Space Systems Simulation Laboratory*. Virginia Polytechnic Institute and State University. <https://vttechworks.lib.vt.edu/handle/10919/84980>
- [66] Kiesbye, J., Messmann, D., Preisinger, M., Reina, G., Nagy, D., Schummer, F., Mostad, M., Kale, T., & Langer, M. (2019). *Hardware-In-The-Loop and Software-In-The-Loop Testing of the MOVE-II CubeSat*. Aerospace, 6(12), 130–. <https://doi.org/10.3390/aerospace6120130>
- [67] da Silva, R. C., Ishioka, I. S. K., Cappelletti, C., Battistini, S., & Borges, R. A. (2019). *Helmholtz Cage Design and Validation for Nanosatellites HWIL Testing*. IEEE Transactions on Aerospace and Electronic Systems, 55(6), 3050–3061. <https://doi.org/10.1109/TAES.2019.2898309>
- [68] Wen, H. Y., Kruizinga, G., Paik, M., Landerer, F., Bertiger, W., Sakumura, C., Bandikova, T., & Mccullough, C. (2019). *Gravity recovery and climate experiment follow-on (GRACE-FO) level-1 data product user handbook*. https://podaac-tools.jpl.nasa.gov/drive/files/allData/gracefo/docs/GRACE-FO_L1_Handbook.pdf
- [69] *satelliteScenario*. MATLAB Aerospace Toolbox Documentation. (2022). Retrieved from <https://www.mathworks.com/help/aerotbx/ug/satellitescenario.html>

- [70] NASA. (n.d.). *Physical Oceanography Distributed Active Archive Center Website*. Retrieved June 11, 2022 from <https://podaac.jpl.nasa.gov/>
- [71] NASA. (n.d.). *Crustal Dynamics Data Information System Website*. Retrieved June 11, 2022 from <https://cddis.nasa.gov/index.html>
- [72] space-track.org. (n.d.). *Historical ELSET Search*. Retrieved June 11, 2022 from <https://www.space-track.org/>
- [73] Flores, A. (2020). *NAVSTAR GPS Space Segment/Navigation User Segment Interfaces*. ICD-GPS-200L. Retrieved October 21, 2021 from <https://www.gps.gov/technical/icwg/>
- [74] Ashby, N. (2004). *The Sagnac Effect in the Global Positioning System*. In: Rizzi, G., Ruggiero, M.L. (eds) Relativity in Rotating Frames. Fundamental Theories of Physics, vol 135. Springer, Dordrecht. https://doi.org/10.1007/978-94-017-0528-8_3
- [75] Ashby, N. & Weiss, M. (1997). *Global Positioning System Receivers and Relativity*. NIST Technical Note 1385. <https://apps.dtic.mil/sti/citations/ADA508049>
- [76] *ecef2eci*. MATLAB Aerospace Toolbox Documentation. (2022). Retrieved from <https://www.mathworks.com/help/aerotbx/ug/ecef2eci.html>
- [77] *eci2ecef*. MATLAB Aerospace Toolbox Documentation. (2022). Retrieved from <https://www.mathworks.com/help/aerotbx/ug/eci2ecef.html>
- [78] Keil, E. M. (2014). *Kalman Filter Implementation to Determine Orbit and Attitude of a Satellite in a Molniya Orbit*. Virginia Polytechnic Institute and State University. <http://hdl.handle.net/10919/49102>
- [79] Chiaradia, A. P. M., Gill, E., Montenbruck, O., Kuga, H.K., & Prado, A. F. B. A., (2000). *Algorithms for On-Board Orbit Determination Using GPS OBODE-GPS*. DLR-Interner Bericht.
- [80] *isoutlier*. MATLAB Documentation. (2022). Retrieved from <https://www.mathworks.com/help/matlab/ref/isoutlier.html>
- [81] Escobal, P. R. (1963). *Rise and Set Time of a Satellite About an Oblate Planet*. AIAA Journal, 1, 2306–2310. <https://doi.org/10.2514/3.2057>
- [82] Palmer, P. & Mai, Y. (2000). *A Fast Prediction Algorithm of Satellite Passes*. 14th Annual AIAA/USU Conference on Small Satellites.
- [83] Han, C., Gao, X., & Sun, X. (2017). *Rapid satellite-to-site visibility determination based on self-adaptive interpolation technique*. Science China. Technological Sciences, 60(2), 264–270. <https://doi.org/10.1007/s11431-016-0513-8>

- [84] Sánchez, M. A., Jilet, B., Setty, S. J., & Flohrer, T. (2019). *Employing Fast Orbit Prediction for Optimisation of Satellite Visibility Computation*. 1st NEO and Debris Detection Conference.
- [85] Wang, X., Han, C., Yang, P., & Sun, X. (2019). *Onboard Satellite Visibility Prediction Using Metamodeling Based Framework*. Aerospace Science and Technology, 94, 105377–. <https://doi.org/10.1016/j.ast.2019.105377>
- [86] Popp, N., Krishnan, V., Vautour, E., Bauer, M. R., Wailand, A., Newton, A., Eastwood, S., Chandrasekaran, S., & Bauer, R. (2021). *Multi-shaft reaction wheel design for a 2U Cubesat*. Progress in Canadian Mechanical Engineering. Volume 4, 4. <https://doi.org/10.32393/csme.2021.34>
- [87] *fminsearch*. MATLAB Documentation. (2022). Retrieved from <https://www.mathworks.com/help/matlab/ref/fminsearch.html>
- [88] Tahoumi, E. (2019). *New Robust Control Schemes Linking Linear and Sliding Mode Approaches*. L'Ecole Centrale De Nantes. <https://tel.archives-ouvertes.fr/tel-02953992>
- [89] Newway Air Bearings. (2007). *Air Bearing Flat Round 040mm Specs*. S104001 datasheet, Oct. 2007.
- [90] Motion Analysis. (2012). *Raptor-4 Specifications*. Rapter-4 datasheet, Oct. 2012.
- [91] Springmann, J.C. (2013). *Satellite Attitude Determination with Low Cost Sensors*. University of Michigan. <http://hdl.handle.net/2027.42/102312>
- [92] ASTM International. (2019). *Standard Classification for Solar Simulators for Electrical Performance Testing of Photovoltaic Devices*. <https://www.astm.org/e0927-19.html>
- [93] *Solar simulation - sun simulators: AM0-AM40: AM1.5G standards*. G2V Optics Inc. (2022, March 22). Retrieved July 26, 2022, from <https://g2voptics.com/solar-simulation/>
- [94] Wang, W., & Laumert, B. (2014). *Simulate a “Sun” for Solar Research : A Literature Review of Solar Simulator Technology*. Retrieved from KTH Royal Institute of Technology website: <http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-154262>
- [95] Leary, G.P. (2016). *Comparison of Xenon Lamp-Based and LED-Based Solar Simulators*. Montana State University. <https://scholarworks.montana.edu/xmlui/handle/1/9837>
- [96] Lisbona, E. F. (2012). Chapter IID-3 - Calibration, Testing, and Monitoring of Space Solar Cells. In A. J. McEvoy, T. Markvart, & Castañer Luis (Eds.), *Practical Handbook of Photovoltaics Fundamentals and Applications* (2nd ed., pp. 865–867). essay, Academic Press.

- [97] ASTM International. (2022). *Standard Solar Constant and Zero Air Mass Solar Spectral Irradiance Tables*. <https://www.astm.org/e0490-22.html>
- [98] Michael, P. R., Johnston, D. E., & Moreno, W. (2022). *A Study on Solar Irradiance and Lux Illuminance: Conversion guide*. Innovations in Science and Technology Vol. 4, 40–56. <https://doi.org/10.9734/bpi/ist/v4/15405d>
- [99] OSRAM Opto Semiconductors. (2018). *DIL SMT Ambient Light Sensor*. SFH2430 datasheet, May. 2018.
- [100] *Nanlite Forza 200 led monolight*. Nanlite US. (n.d.). Retrieved July 26, 2022, from <https://nanliteus.com/nanlite-forza-200-led-monolight/>

Appendix A. Algorithms

Algorithm 3.1: GPS Sensor Model

for each satellite i of n tracked satellites:

1. calculate Δt_{s_i} according to Section 20.3.3.3.1 of the GPS ICD

2. $\rho_{s_i}' = \rho_{s_i} + c \times b_{s_i}$

3. calculate \mathbf{r}_{s_i} according to 20.3.3.4.3 of the GPS ICD

4. $\theta = \omega_{\oplus} \frac{\rho_{s_i}}{c}$

5. $\mathbf{r}_{s_i}' = \mathbf{R}_z(\theta) \mathbf{r}_{s_i} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{r}_{s_i}$

$$\boldsymbol{\rho} = [\rho_{s_1}' \quad \rho_{s_2}' \quad \cdots \quad \rho_{s_n}']^T$$

$$\mathbf{r}_s = [\mathbf{r}_{s_1}' \quad \mathbf{r}_{s_2}' \quad \cdots \quad \mathbf{r}_{s_n}']^T$$

Algorithm 3.2: GPS SPS

$$|\Delta\hat{\mathbf{r}}| = 1$$

while $|\Delta\hat{\mathbf{r}}| > 1E - 3$

1. for each satellite i of n tracked satellites:

$$a. \quad \hat{\rho}_{s_i} = |\mathbf{r}_{s_i} - \hat{\mathbf{r}}| + c\hat{b}$$

$$b. \quad \Delta\rho_{s_i} = \hat{\rho}_{s_i} - \rho_{s_i}$$

$$c. \quad \widehat{\mathbf{1}}_i^T = \frac{\mathbf{r}_{s_i} - \hat{\mathbf{r}}}{|\mathbf{r}_{s_i} - \hat{\mathbf{r}}|}$$

$$2. \quad \Delta\boldsymbol{\rho} = \begin{bmatrix} \Delta\rho_{s_1} \\ \Delta\rho_{s_2} \\ \vdots \\ \Delta\rho_{s_n} \end{bmatrix} \quad \& \quad \mathbf{G} = \begin{bmatrix} -\widehat{\mathbf{1}}_1^T & 1 \\ -\widehat{\mathbf{1}}_2^T & 1 \\ \vdots & \vdots \\ -\widehat{\mathbf{1}}_n^T & 1 \end{bmatrix}$$

$$3. \quad \Delta\hat{\mathbf{x}} = \begin{bmatrix} \Delta\hat{\mathbf{r}} \\ c\Delta\hat{b} \end{bmatrix} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \Delta\boldsymbol{\rho}$$

$$4. \quad \hat{\mathbf{r}} = \hat{\mathbf{r}} - \Delta\hat{\mathbf{r}}$$

$$5. \quad \hat{b} = \hat{b} - \Delta\hat{b}$$

end

$$\hat{\mathbf{v}} = \frac{\hat{\mathbf{r}} - \hat{\mathbf{r}}_0}{\Delta t}$$

$$\hat{\mathbf{f}} = \frac{\hat{b} - \hat{b}_0}{\Delta t}$$

Algorithm 3.3: RDEKF prediction phase

1. convert satellite states into ECI coordinates
 2. $\widehat{\mathbf{x}}_{c_k}^- = \boldsymbol{\phi}_c \widehat{\mathbf{x}}_{c_{k-1}}$
 3. $\widehat{\mathbf{x}}_{p_k}^- = \text{predicted using Cowell's method}$
 4. a) $\mathbf{F} = \frac{\partial \dot{\mathbf{x}}_p}{\partial \mathbf{x}_p}$
b) $\boldsymbol{\Phi}_p \approx \mathbf{I} + \mathbf{F}\Delta t + \frac{1}{2!}\mathbf{F}^2\Delta t^2$
 5. a) $\widehat{\mathbf{x}}_k^- = \begin{bmatrix} \widehat{\mathbf{x}}_{p_k}^- \\ \widehat{\mathbf{x}}_{c_k}^- \end{bmatrix}$
b) $\boldsymbol{\phi} = \begin{bmatrix} \boldsymbol{\phi}_p & 0 \\ 0 & \boldsymbol{\phi}_c \end{bmatrix}$
 6. $\mathbf{p}_k^- = \boldsymbol{\phi} \mathbf{p}_{k-1} \boldsymbol{\phi}^T + \mathbf{Q}$
 7. convert satellite states back into ECEF coordinates
-

Algorithm 3.4: RDEKF update phase

1. for each satellite i of n tracked satellites:
 - a. $\hat{\rho}_{s_i} = |\mathbf{r}_{s_i} - \widehat{\mathbf{r}}_k^-| + c\widehat{b}_k^-$
 - b. $\Delta\rho_{s_i} = \rho_{s_i} - \hat{\rho}_{s_i}$
 - c. $\widehat{\mathbf{1}}_i^T = \frac{\mathbf{r}_{s_i} - \widehat{\mathbf{r}}_k^-}{|\mathbf{r}_{s_i} - \widehat{\mathbf{r}}_k^-|}$
 2. $\Delta\boldsymbol{\rho} = \begin{bmatrix} \Delta\rho_{s_1} \\ \Delta\rho_{s_2} \\ \vdots \\ \Delta\rho_{s_n} \end{bmatrix}$ & $\mathbf{H} = \begin{bmatrix} -\widehat{\mathbf{1}}_1^T & 0 & 0 & 0 & 1 & 0 \\ -\widehat{\mathbf{1}}_2^T & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\widehat{\mathbf{1}}_n^T & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
 3. $\mathbf{K} = \mathbf{p}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{p}_k^- \mathbf{H}^T + \mathbf{R})^{-1}$
 4. $\widehat{\mathbf{x}}_k^+ = \widehat{\mathbf{x}}_k^- + \mathbf{K} \Delta\boldsymbol{\rho}$
 5. $\mathbf{p}_k^+ = (\mathbf{I} - \mathbf{K} \mathbf{H}) \mathbf{K} \mathbf{H} (\mathbf{I} - \mathbf{K} \mathbf{H})^T + \mathbf{K} \mathbf{R} \mathbf{K}^T$
-

Algorithm 4.1: Palmer and Mai Coarse Target Prediction

```
if  $\phi_s > \phi_T + \theta_T$ 
     $\Delta\phi_v = \phi_s - (\phi_T + \theta_T)$ 
else
     $\Delta\phi_v = 360 + \phi_s - (\phi_T + \theta_T)$ 
end
```

$$N = \left\lceil \frac{\Delta\phi_v}{\hat{P}_s \omega_{\oplus}} \right\rceil + 1$$

Algorithm 4.2: Modified Coarse Target Prediction

```
if  $\phi_s > \phi_T + \theta_T$ 
     $\Delta\phi_v = \phi_s - (\phi_T + \theta_T)$ 
else if  $\phi_s - \hat{P}_s \omega_{\oplus} > \phi_T - \theta_T$ 
     $\Delta\phi_v = 1$ 
else
     $\Delta\phi_v = 360 + \phi_s - (\phi_T + \theta_T)$ 
end
```

$$N = \left\lceil \frac{\Delta\phi_v}{\hat{P}_s \omega_{\oplus}} \right\rceil + 1$$

Algorithm 4.3: Fine Target Prediction

```
 $\Delta\lambda = \lambda_s - \lambda_T$ 
 $\Delta\phi = \phi_s - \phi_T$ 
if  $\Delta\lambda < \theta_T$ 
     $\Delta\lambda' = \Delta\lambda$ 
     $\Delta\phi' = \Delta\phi - \hat{P}_s \omega_{\oplus}$ 
     $\Delta\theta' = (\Delta\lambda'^2 + \Delta\phi'^2)^{0.5}$ 
    if  $\Delta\theta' < \theta_T$ 
         $t' = t + \hat{P}_s$ 
        return  $t'$ 
    end
end
```

Appendix B. Flowcharts

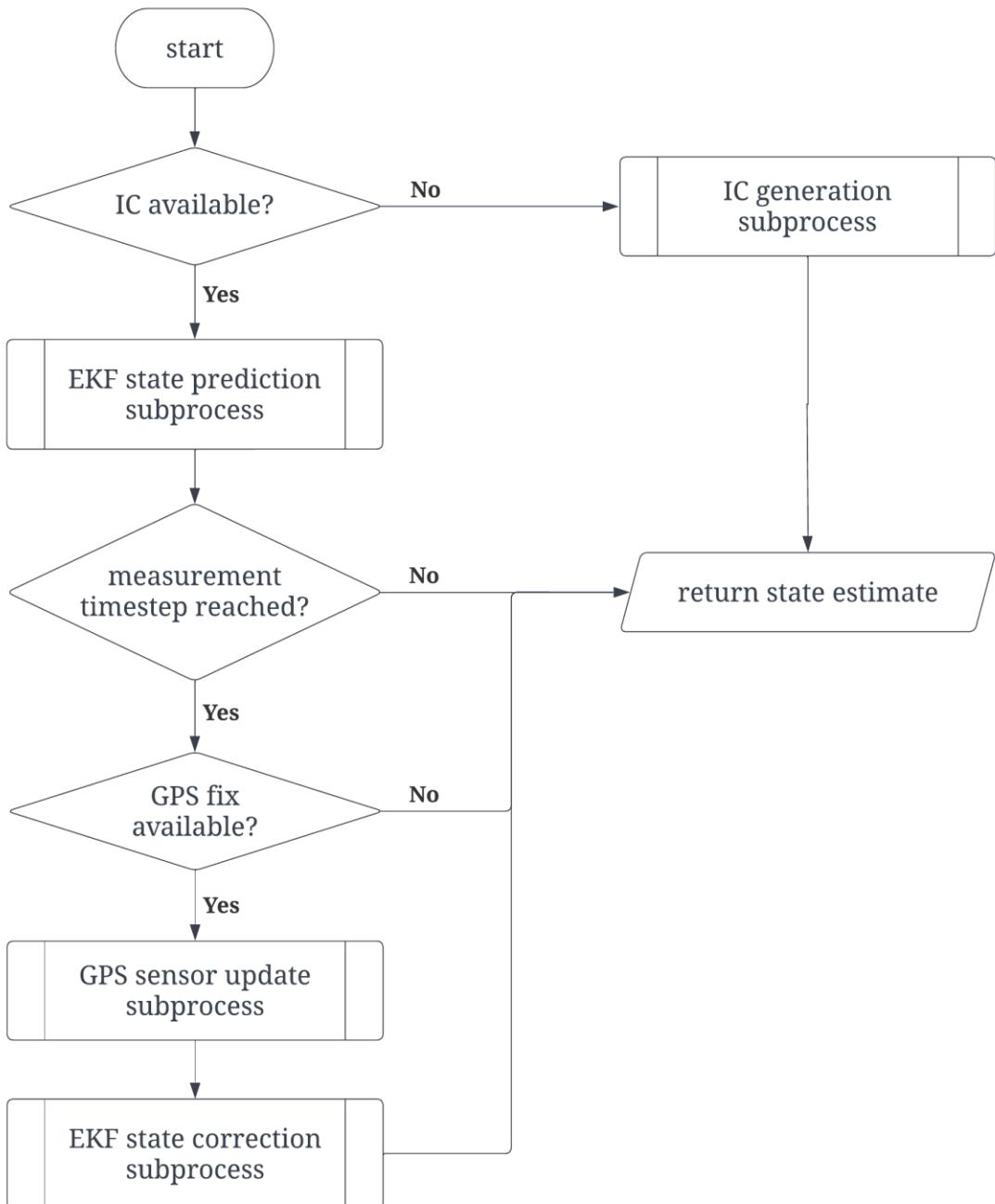


Figure 38. Reduced dynamic extended Kalman filter for GPS-based onboard orbit determination.

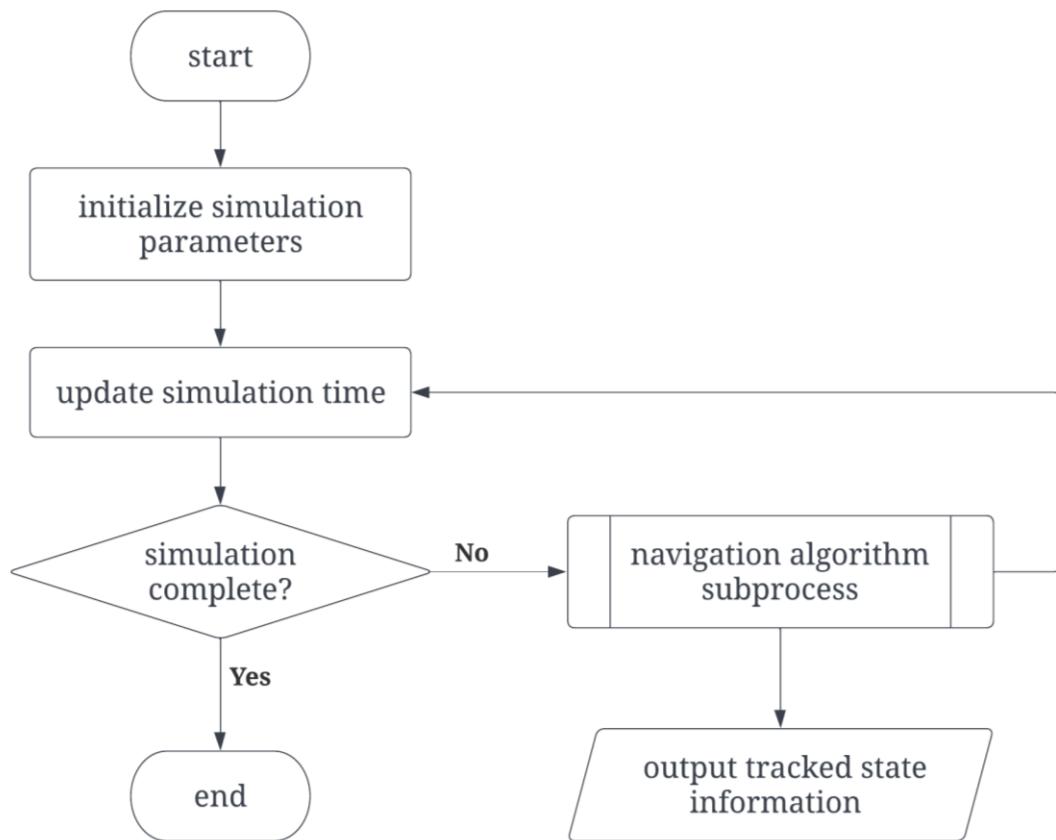


Figure 39. ODS simulator process flowchart.

Appendix C. Additional Results

C.1 RDEKF Sensitivity Analysis Testing

Table 13. RDEKF sensitivity analysis testing results, part 1: 3D position RMSE vs. measurement time step

measurement timestep [s]	duty cycle [%]	RMSE [m]		maximum error [m]	
		TB	TBJ2	TB	TBJ2
10	100.0	20.59	19.75	61.88	60.93
20	100.0	22.72	20.12	63.46	58.80
30	100.0	26.84	20.45	75.23	62.20
60	100.0	52.66	21.05	159.55	67.26
90	100.0	97.89	22.18	293.09	91.64
180	100.0	324.85	25.41	979.97	164.75
270	66.7	661.41	33.06	2016.08	236.96
360	50.0	1074.72	43.68	3367.70	307.28
450	40.0	1532.95	55.00	4907.03	374.75
540	33.3	-	70.06	-	438.45
630	28.6	-	86.59	-	497.53
720	25.0	-	99.46	-	551.38
810	22.2	-	116.45	-	599.96
900	20.0	-	134.13	-	643.91
990	18.2	-	154.85	-	684.47
1080	16.7	-	164.12	-	723.58
1170	15.4	-	199.73	-	764.16
1260	14.3	-	217.39	-	810.19
1350	13.3	-	245.79	-	866.47
1440	12.5	-	278.37	-	937.96
1530	11.8	-	342.13	-	1029.01
1620	11.1	-	383.86	-	1142.51
1710	10.5	-	478.47	-	1280.30
1800	10.0	-	566.61	-	1443.46

Table 14. RDEKF sensitivity analysis testing results, part 2: 3D velocity RMSE vs. measurement time step

measurement timestep [s]	duty cycle [%]	RMSE [m/s]		maximum error [m/s]	
		TB	TBJ2	TB	TBJ2
10	100.0	0.481	0.286	5.502	5.503
20	100.0	0.618	0.238	5.502	5.503
30	100.0	0.791	0.228	5.502	5.503
60	100.0	1.327	0.187	5.502	5.503
90	100.0	1.867	0.174	5.502	5.503
180	100.0	3.392	0.173	7.310	5.503
270	66.7	4.714	0.188	10.455	5.503
360	50.0	5.815	0.205	13.222	5.503
450	40.0	6.705	0.218	15.507	5.503
540	33.3	-	0.232	-	5.503
630	28.6	-	0.245	-	5.503
720	25.0	-	0.253	-	5.503
810	22.2	-	0.262	-	5.503
900	20.0	-	0.270	-	5.503
990	18.2	-	0.283	-	5.503
1080	16.7	-	0.279	-	5.503
1170	15.4	-	0.313	-	5.503
1260	14.3	-	0.324	-	5.503
1350	13.3	-	0.348	-	5.503
1440	12.5	-	0.378	-	5.503
1530	11.8	-	0.445	-	5.503
1620	11.1	-	0.491	-	5.503
1710	10.5	-	0.591	-	5.503
1800	10.0	-	0.675	-	5.503

C.2 OGNS Testing

C.2.1 Target Flyby Estimation Error: Position Error Plots for Day 226 of 2018

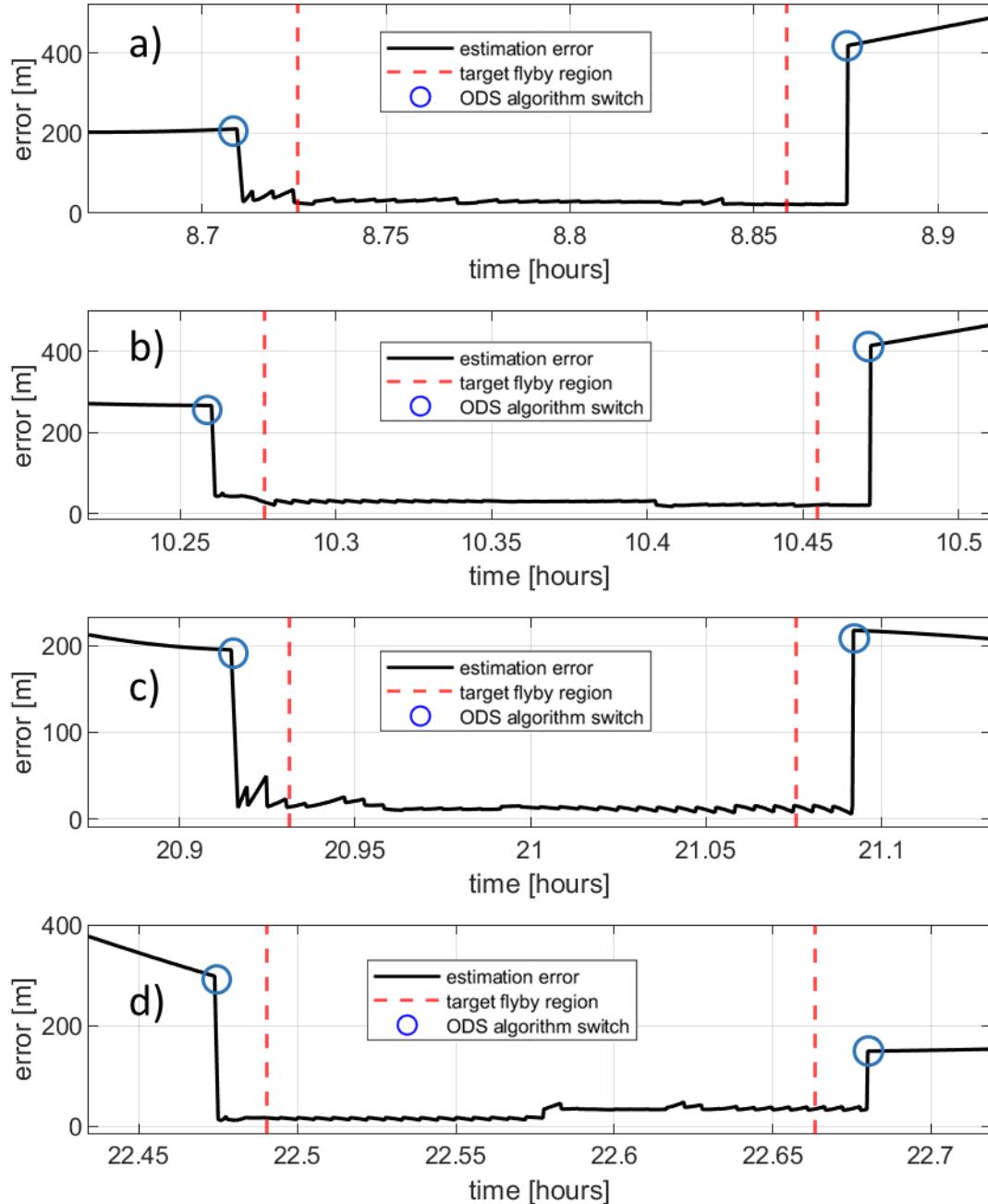


Figure 40. OGNS 3D position error vs. time for Halifax target flybys on Day 226 of 2018. Plots a) – d) correspond to four separate flybys which were all successfully identified by the OGNS algorithm.

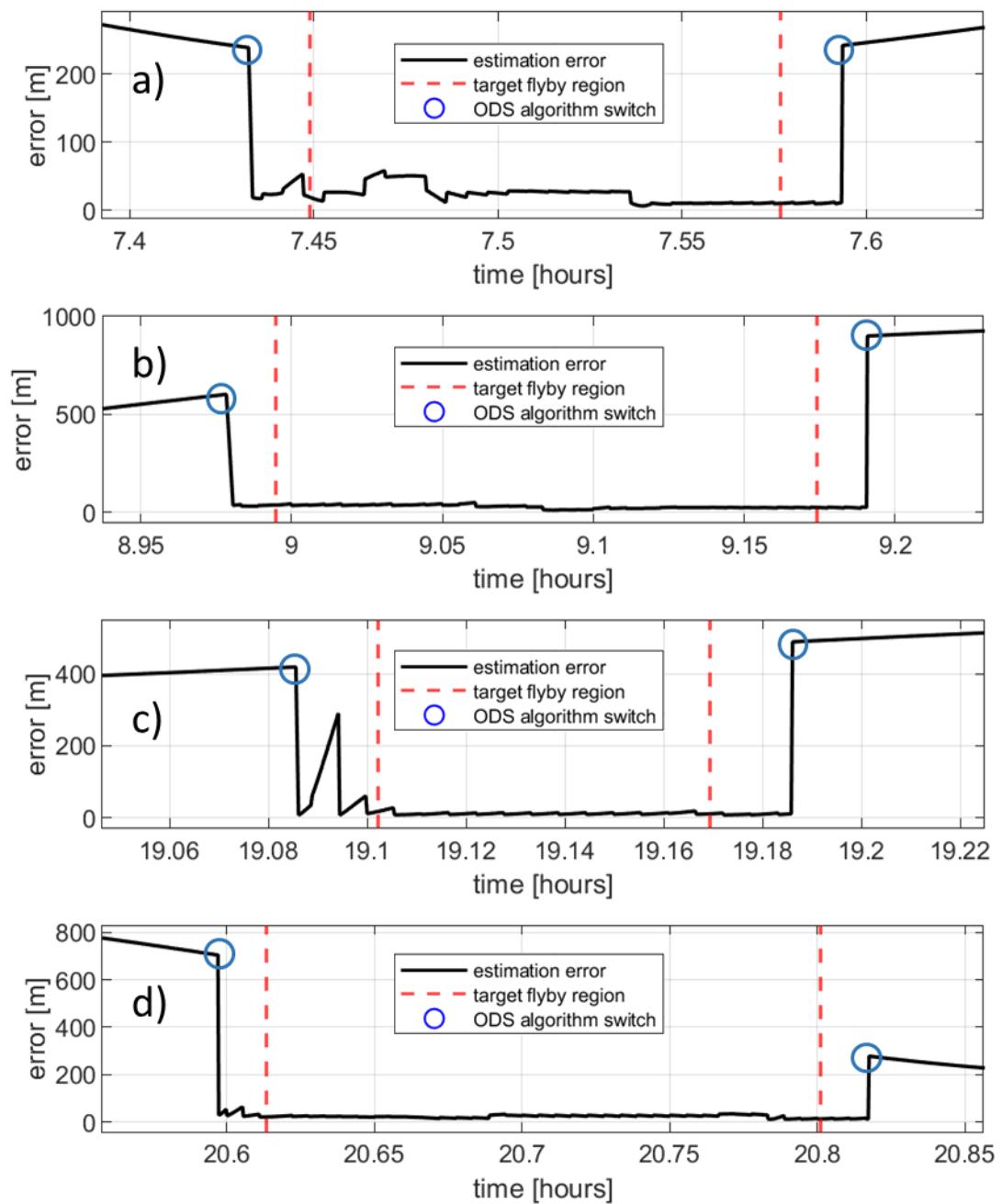


Figure 41. OGNS 3D position error vs. time for Rio target flybys on Day 226 of 2018. Plots a) – d) correspond to four separate flybys which were all successfully identified by the OGNS algorithm.

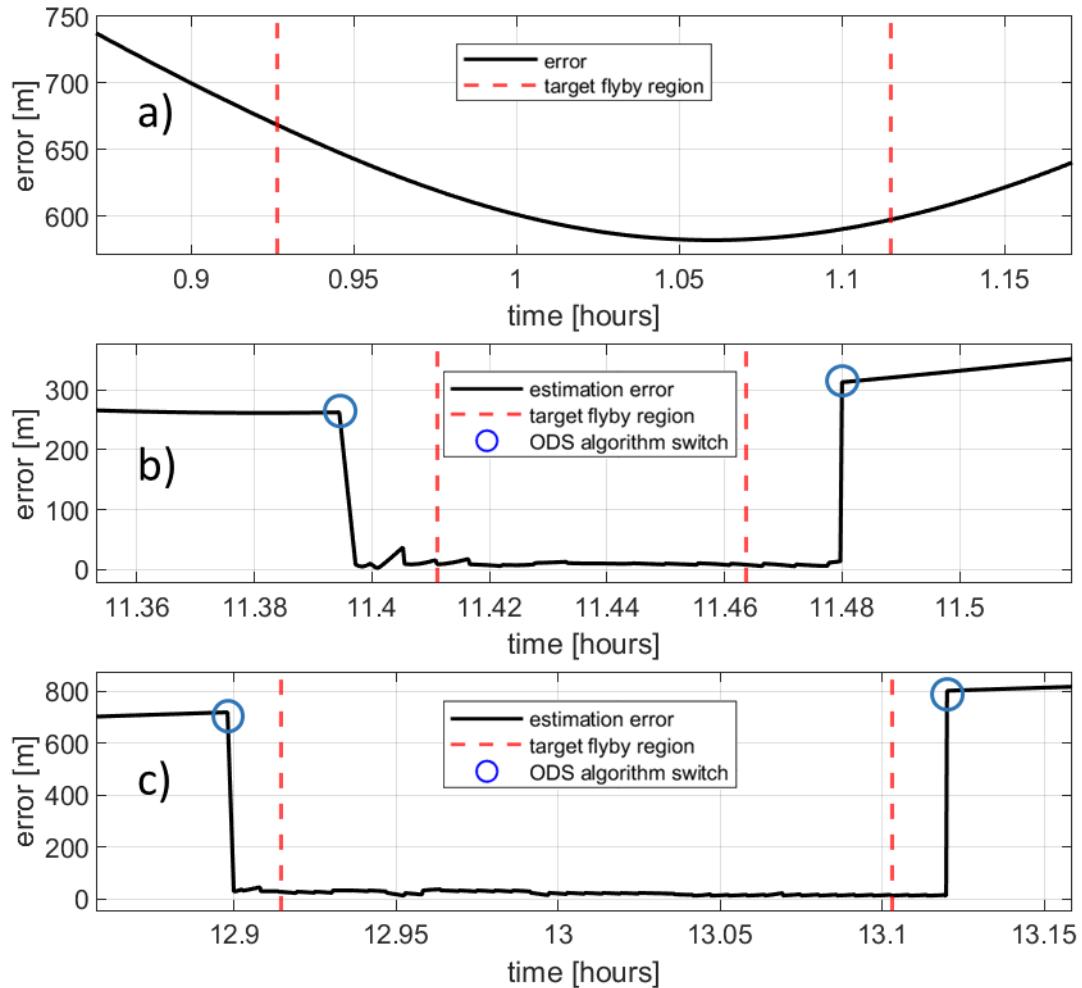


Figure 42. OGNS 3D position error vs. time for Mumbai target flybys on Day 226 of 2018. Plots a) – c) correspond to three separate flybys. The first flyby in plot a) was missed by the OGNS algorithm due to insufficient information at start up; however, the remaining flybys were successfully identified by the OGNS algorithm.

C.2.2 Target Flyby Estimation Error: Results Summary for Days 226 – 229 of 2018

Table 15. 96-hour Halifax target flyby OGNS testing results summary (NOTE: green cells indicate successful flybys with no outliers, yellow cells indicate successful flybys with outliers that have been removed, red cells indicate flybys which were missed and were therefore not included in the average results)

	flyby duration [min]	3D position error		3D velocity error	
		RMSE [m]	max [m]	RMSE [m/s]	max [m/s]
flyby 1	7.70	28.47	36.86	0.43	1.17
flyby 2	10.67	27.54	32.59	0.39	1.10
flyby 3	8.65	13.55	25.57	0.33	0.59
flyby 4	10.37	27.52	48.06	0.39	1.10
flyby 5	4.52	38.23	49.34	0.42	0.91
flyby 6	11.30	43.28	50.93	0.41	0.98
flyby 7	6.12	21.04	29.75	0.33	0.55
flyby 8	11.15	41.46	66.73	0.57	1.55
flyby 9	11.50	48.77	773.71	0.48	4.51
flyby 10	11.45	55.51	795.37	0.63	3.32
flyby 11	11.23	25.09	45.33	0.45	1.38
flyby 12	5.82	20.89	28.32	0.39	0.79
flyby 13	11.30	27.18	41.48	0.37	0.77
flyby 14	4.02	10.47	11.86	0.33	0.50
average	8.99	30.64	145.42	0.42	1.37

Table 16. 96-hour Rio target flyby OGNS testing results summary (NOTE: green cells indicate successful flybys with no outliers, yellow cells indicate successful flybys with outliers that have been removed, red cells indicate flybys which were missed and were therefore not included in the average results)

	flyby duration [min]	3D position error		3D velocity error	
		RMSE [m]	max [m]	RMSE [m/s]	max [m/s]
flyby 1	7.67	27.17	58.30	0.43	1.11
flyby 2	10.77	31.44	51.12	0.35	0.72
flyby 3	4.03	12.56	27.82	0.34	1.00
flyby 4	11.27	23.84	33.64	0.37	1.48
flyby 5	4.07	37.81	47.12	0.35	0.85
flyby 6	11.32	31.49	48.56	0.31	0.68
flyby 7	11.23	23.94	42.30	0.56	3.60
flyby 8	11.43	24.03	38.71	0.37	0.86
flyby 9	11.07	22.32	41.16	0.53	3.23
flyby 10	6.10	19.74	36.65	0.36	0.81
flyby 11	11.10	16.36	35.18	0.37	1.10
flyby 12	6.13	34.08	69.48	0.81	3.61
flyby 13	10.27	19.17	39.75	0.40	1.33
flyby 14	8.65	918.96	1077.55	2.36	3.39
average	8.96	24.92	43.83	0.43	1.57

Table 17. 96-hour Mumbai target flyby OGNS testing results summary (NOTE: green cells indicate successful flybys with no outliers, yellow cells indicate successful flybys with outliers that have been removed, red cells indicate flybys which were missed and were therefore not included in the average results)

	flyby duration [min]	3D position error		3D velocity error	
		RMSE [m]	max [m]	RMSE [m/s]	max [m/s]
flyby 1	11.33	605.52	668.66	2.79	4.06
flyby 2	2.82	9.61	13.15	0.28	0.48
flyby 3	11.30	22.86	37.05	0.35	0.71
flyby 4	11.42	18.27	38.33	0.53	3.60
flyby 5	11.33	16.25	33.26	0.45	1.43
flyby 6	11.07	26.26	42.00	0.28	0.68
flyby 7	6.32	10.92	18.03	0.31	0.99
flyby 8	10.98	10.99	24.67	0.32	0.78
flyby 9	6.55	19.09	51.53	0.48	1.26
flyby 10	10.25	22.94	44.69	0.32	0.82
flyby 11	8.78	13.67	22.05	0.31	0.82
flyby 12	10.08	10.87	24.02	0.37	0.98
flyby 13	8.92	16.81	37.12	0.32	0.75
flyby 14	8.80	34.81	56.37	0.38	0.93
average	9.12	17.95	34.02	0.36	1.09

C.3 ADCS Device Testing

C.3.1 RWA Negative Step Input Test Results

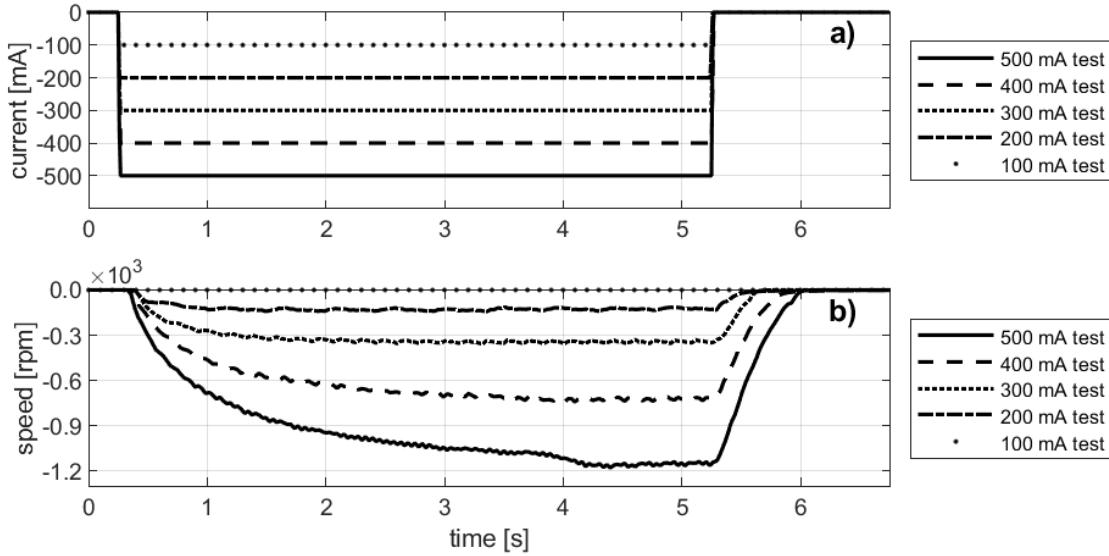


Figure 43 RWA negative step input test summary plots for set point currents within the motors nominal operating range: a) input current and b) output speed vs. time. The speed response is approximately first order suggesting significant damping in the system.

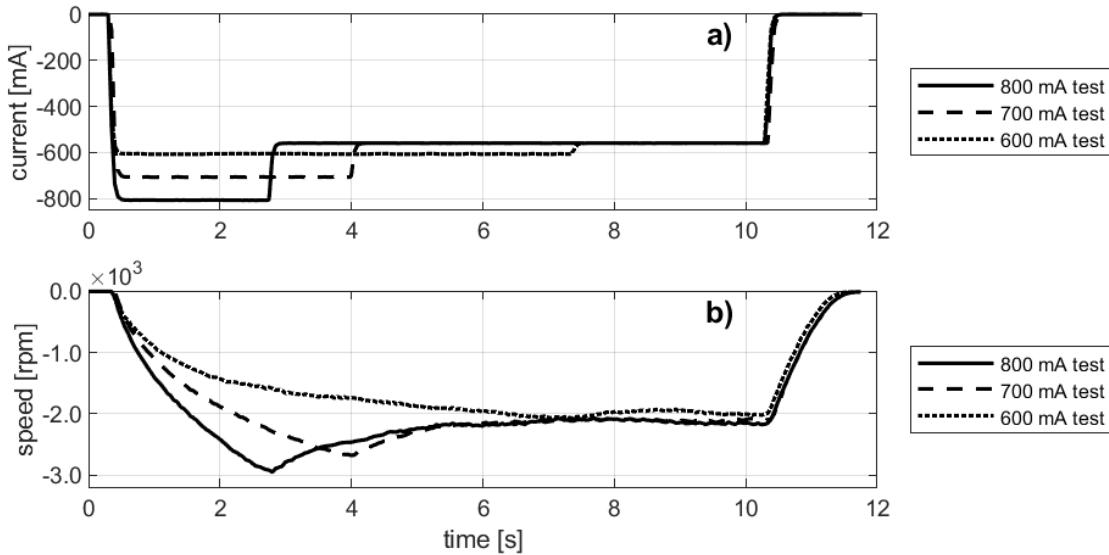


Figure 44. RWA negative step input test summary plots for set point currents outside the motors nominal operating range: a) input current and b) output speed vs. time. The maximum steady-state speed reached by the RWA is increased when using set point currents above the motors nominal operating range; however, the motor controller automatically limits the applied current after a set time period to prevent overheating.

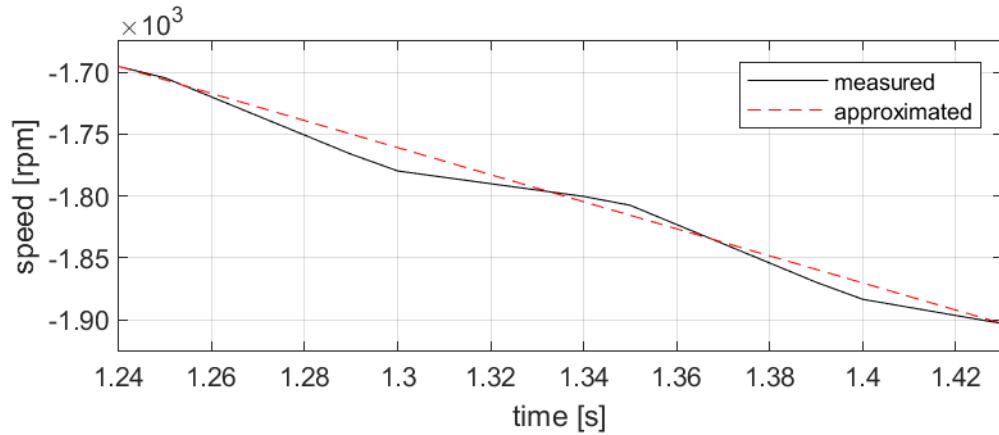


Figure 45. RWA constant acceleration speed response compared to experimental measurements for the negative 800 mA step input test. As expected, the constant acceleration approximation closely matches the experimental response over small speed ranges.

C.3.2 RSS Slew Controller Testing

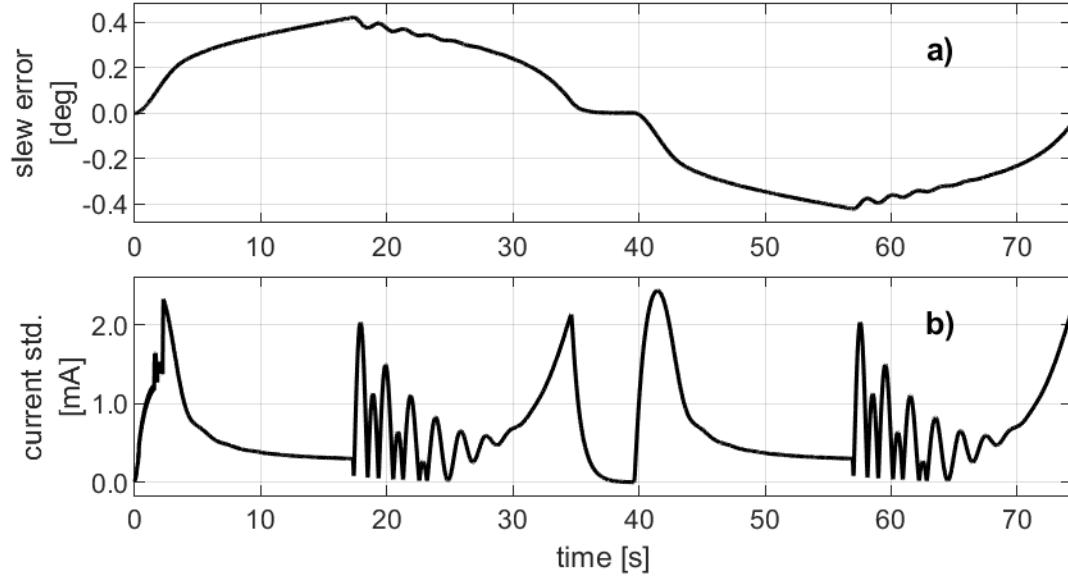


Figure 46. RSS slew controller simulation tuning test 4 results summary: a) slew error and b) moving current standard deviation vs. time. This test used a proportional gain of 1000 mA/deg and resulted in the best performance in simulation.

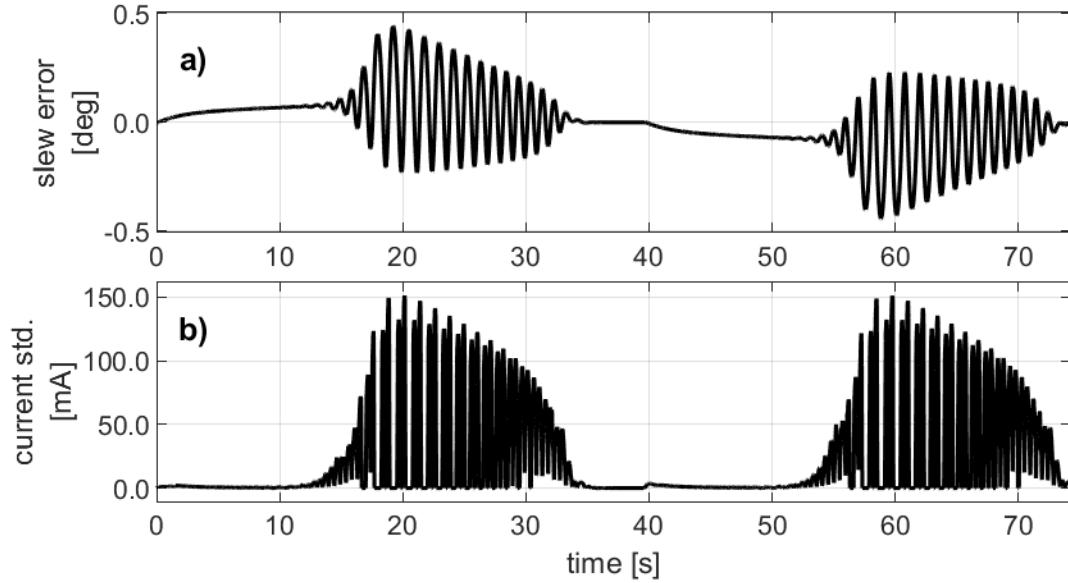


Figure 47. RSS slew controller simulation tuning test 5 results summary: a) slew error and b) moving current standard deviation vs. time. This test used a proportional gain of 5000 mA/deg and resulted in the loss of controller stability as indicated by the significant increase in the current standard deviation compared to test 4 in Figure 46 as well as the high frequency oscillations seen in both plots a) and b) here.

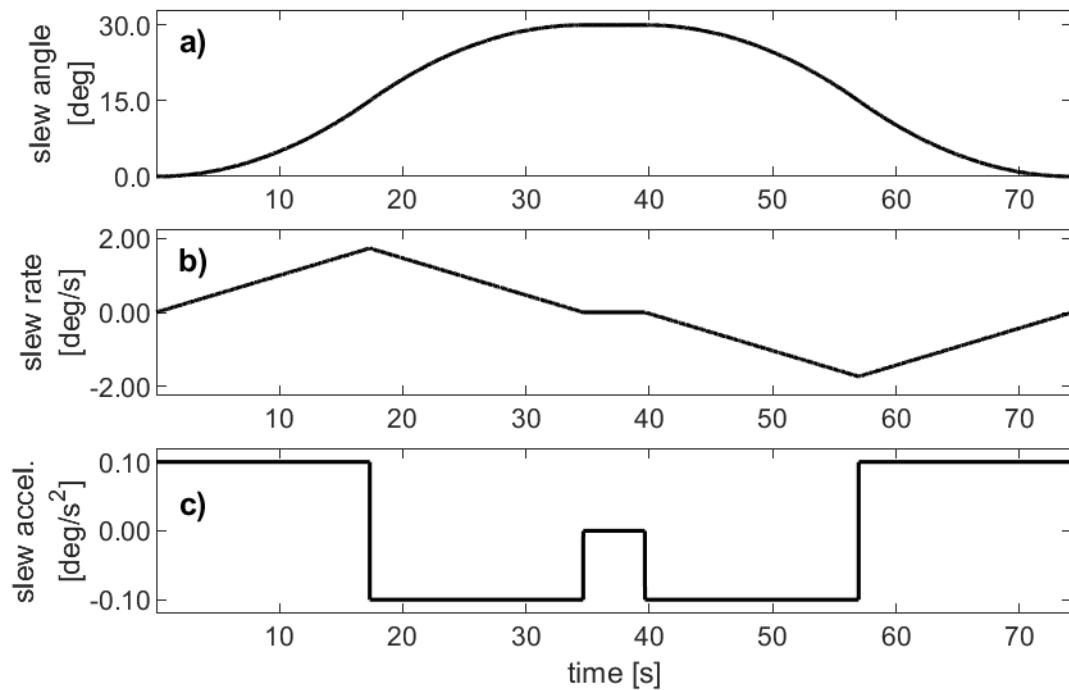


Figure 48. 30-degree slew trajectory for testing the RSS slew controller: a) slew angle, b) slew rate, and c) slew acceleration vs. time.

C.3.3 Sun Sensor Testing

Table 18. Sun sensor intensity threshold sensitivity analysis results

threshold [%]	max error [deg]	average error [deg]	error std. [deg]	error RMSE [deg]	dropouts [%]
0	64.45	-19.27	20.15	27.88	0.00
1	64.45	-19.27	20.15	27.88	0.00
2	64.45	-19.27	20.15	27.88	0.00
3	64.45	-19.27	20.15	27.88	0.00
4	46.93	-0.87	17.45	17.47	0.00
5	53.41	-7.56	16.81	18.43	0.00
6	53.41	-7.87	16.52	18.25	0.65
7	59.15	-8.10	16.54	18.32	1.26
8	64.59	-8.34	16.66	18.48	1.97
9	69.15	-8.60	16.88	18.73	2.75
10	73.67	-8.87	17.25	19.11	3.66
11	76.95	-9.20	17.77	19.63	4.78
12	80.08	-9.58	18.51	20.35	5.91
13	83.11	-9.91	19.48	21.23	7.15
14	85.53	-9.86	20.60	22.04	8.36
15	89.03	-9.57	22.16	23.15	9.47
16	89.24	-9.37	23.32	23.96	10.54
17	96.31	-12.74	25.03	26.75	11.63
18	95.45	-12.24	26.53	27.63	12.79
19	93.30	-11.81	27.53	28.08	14.33
20	92.88	-11.32	28.60	28.53	16.08
21	91.72	-11.55	29.63	29.22	17.94
22	81.96	-3.00	30.57	27.61	19.40
23	81.22	-2.53	31.67	28.28	20.90
24	80.12	-2.07	32.82	28.93	22.68
25	80.06	-1.54	34.08	29.56	24.97
26	81.63	-1.05	35.39	30.19	27.31
27	1519.13	502.59	510.54	660.08	29.73
28	3316.79	1463.11	1071.19	1706.78	32.67
29	4035.17	1865.25	1323.62	2146.32	35.63
30	4393.73	1937.66	1430.19	2237.64	38.75