Prox-1

Guidance, Navigation & Control Formulation and Algorithms

Richard Zappulla II

Image Credit: NASA voyager.jpl.nasa.gov

THIS PAGE IS INTENTIONALY LEFT BLANK

Abstract

Beginning with the manned Gemini missions, proximity operations and rendezvous between two (2) spacecraft have significantly evolved from human-in-the-loop to ground-in-the-loop to more autonomous vehicles, such as the Japanese ETS-VII and the Russian Progress vehicles. Prior to the proposal for the Prox-1 mission, numerous other missions—such as XSS-10, XSS-11, DART, and Orbital Express—have demonstrated varying levels of autonomy. Unlike previous missions, the Prox-1 mission will utilize a completely autonomous GN&C system driven by an on-board GPS receiver, an uncooled infrared microbolometer, a three-axis magnetometer, an inertial measurement unit (IMU), and sun sensors. The GN&C algorithms and strategies discussed in this paper are designed around robust formulations that are shown to guarantee asymptotic stability and aid in mitigating risk involved with passive, autonomous proximity operations.

Acknowledgements

The past several years at this institution has been a dream come true. With each passing year, the experience became more fruitful, enriching, and constantly redefined the expectation of the next year; the people I have met, the places I have been to, and the activities and opportunities that I have had the privilege to be a part of have exceed my wildest expectations. While my time at Georgia Tech is relatively small compared to others, I could not have done without the support from others.

First and foremost I would like to thank my parents Barbara and Richard Zappula for all of the love and support they have given me throughout my (albeit short) career at this Institute; I would also like to thank my extended family, John, Sandy, and Cristina Beltrani, Michael Cici, Pauline, Angelo, (Little) Angleo, Billy, Lisa, and Nikki Gionis for always being there for me as well. I also could not have been as successful as I was without the support of my peers, especially (in no particular order): Alex Buchanan, Kevin Reilley, Ryan Cornell, Michael Mullen, Rick Sigman, and Alex Trimm. Lastly, I would like to thank **ALL** of the past and present members of the Prox-1 GN&C team, especially (and again, in no particular order): Sean Chait, Nishant Prasadh, and Alfredo Valverde for all the hard work, dedication, and all-nighters they have put into the Prox-1 GN&C system. Lastly, I would like to thank Dr. Josue Muñoz, Dr. Frank Chavez, Dr. David Voss, Mr. Nathan Stastny, and Dr. Eric Feron for all of the technical guidance and support they have given not only myself but also the Prox-1 GN&C team to help our mission succeed. I would like to thank my advisor, Professor David Spencer for his assistance and guidance over the past two and a half years. Lastly, I would like to thank the SMART Scholarship program office for providing with this opportunity to expand my knowledge and skill base through this terrific program.

Quest oculus non vide, cor non delet
What the eye does not see, the heart does not feel
"A lot happens that we are not telling you about"
-Opening lines of Apollo software source code

Acronyms

A/D	Attitude Determination
AVGS	Advanced Video Guidance Sensor
b	bit
B	byte
BF	Body-Fixed
BFCF	Body-Fixed Coordinate Frame
BLOB	Binary Large OBject
COB	Center of Brightness
COTS	Commercial Off-The-Shelf
CW	Clohessy-Wiltshire
DART	Demostration of Autonomous Rendezvous Technology
DGPS	Differntial GPS
DSM	Design System Matrix
ECEF	Earth Centered Earth Fixed
EKF	Extended Kalman Filter
FPA	Focal Plane Array
GMST	Greenwich Mean Sidereal Time
GMT	Greenwich Mean Time
GN&C	Guidnace Navigation & Control
GPS	Global Positioning System
ICF	Image Coordinate Frame
IERS	Internation Earth Rotational and Referece System Service
iid	independent and identically distributed
IMU	Inertial Measurement Unit
IPAs	Image Processing Algorithms
ISS	International Space Station
kb	kilobit
kB	kilobyte
KF	Kalman Filter
km	kilometer
LCF	Lyapunov Candidate Function
Mb	megabit
MB	megabyte
MP	Mega Pixel
MTI	Microbolometer Thermal Imager
NMC	Natural Motion Circumnavigation
O/D	Orbtial Determination
RGB	Red-Green-Blue
RHR	Right Hand Rule
RSO	Resident Space Object
RSW	Satellite Coordinate Frame
SRP	Solar Radiation Pressure
STC	Slew and Tracking Controller
STM	State Transition Matrix
TOF	Time of Flight
VC	Visable Camera

Table of Contents

Table of Figures

Table of Tables

1. Introduction

1.1. *Initial Forays into Spacecraft GN&C*

Beginning with the manned Gemini missions, proximity operations and rendezvous between two (2) spacecraft have significantly evolved from human-in-the-loop to ground-in-the-loop to more autonomous vehicles, such as the Japanese ETS-VII and the Russian Progress vehicles, both of which are capable of autonomous docking with the International Space Station (ISS) [1-4]. The difficulties of both proximity operations and formation flying were first encountered during the Gemini program, where pilots attempted to rendezvous with various objects in orbit (such as spent booster sections and other spacecraft) and achieved varying levels of success. While astronauts Gus Grissom and John Young were able to successfully maneuver their capsule via dead reckoning on Gemini III, the Gemini IV astronauts found great difficulty attempting to perform proximity operations. As a result, the first set of guidance algorithms were developed for an on-board digital computer manufactured by IBM that had only 4,096 words of 39-bit memory. The computer calculated trajectories and provided the astronauts with thrust and orientation suggestions. However, as the missions grew more complex and the need for more accurate and efficient celestial navigation increased, the desire for more automated guidance, navigation, and control (GN&C) algorithms grew. Some of these improved algorithms included the inclusion of the Kalman Filter to produce more accurate navigation solutions during the Apollo program [1].

1.2. *Steps Towards Autonomy*

In the present day, while the basics of GN&C algorithms remain fundamentally unchanged from the late 1960's, the capabilities of spacecraft have increased exponentially. This increased capability is primarily due to the increase in sophistication of on-board computational hardware and software. This increased capability of on-board computational hardware allowed for development in the area of automation, among many others.

Prior to the proposal for the Prox-1 mission, numerous other missions—such as XSS-10, XSS-11, DART, and Orbital Express—have demonstrated varying levels of autonomy enabled by varying levels of technology [5-7]. The Demonstration of Autonomous Rendezvous Technology (DART) mission was designed to demonstrate the use of various autonomous rendezvous capabilities, including the Advanced Video Guidance Sensor (AVGS). DART was to perform autonomous rendezvous and proximity operations about a communications satellite that was specially outfitted with fiducial markers that aided the DART spacecraft with its image-based guidance scheme. The XSS-10 and XSS-11 microsatellites demonstrated autonomous proximity operations via visual cameras and LIDAR systems, respectively.

1.3. *The Prox-1 Mission: Automated GGN&C*

Unlike previous missions, the Prox-1 mission will utilize a completely autonomous GN&C system driven by an on-board GPS receiver, an uncooled infrared microbolometer, a three-axis magnetometer, an inertial measurement unit (IMU), and sun sensors. Inertial navigations are obtained via GPS and an attitude determination filter that utilizes IMU, sun sensor, and magnetometer measurements. Relative navigation solutions (i.e. relative position) are driven primarily by the range and angles estimates produced via image processing algorithms (IPAs) applied to infrared images from the microbolometer; inputs from the inertial attitude solution are also used. Spacecraft guidance is provided via Artificial Potential Functions (APFs), algorithms that are driven by relative position estimates. In order to track the guidance solutions, the Prox-1 spacecraft will utilize the TORC Box90 control moment gyro (CMG) array along with a hydrazine thruster from Stellar Exploration.

An autonomous GN&C system such as that of Prox-1 requires a great deal of interconnectivity among its various components. Figure 1 illustrates at a high-level the first-order connections between the various components of the GN&C system, along with the critical inputs to and from each element. Additionally, each of subset of the GN&C systems are color-coded such that all Guidance blocks are green, all Navigation blocks are blue, and all Control blocks are light orange.

Figure 2 illustrates the general automated sequence that Prox-1 will follow. Once the ground station authorizes Prox-1 to enter proximity operations, the spacecraft will first need to locate the Resident Space Object (RSO) by using an estimate of the last known point (LKP) of the RSO. If the RSO is not located in the field of view (FOV), Prox-1 will then begin a raster scan using the initial error covariance for the Relative Orbital Determination (O/D) filter – shown in Table 6. This error covariance forms an ellipsoidal projection around the RSO such that the bounds of the ellipsoid are defined to be the 3σ error bounds. Once the target is detected to reside inside of an image, a Relative O/D solution is obtained. Once the filter converges, the relative position estimate is then passed to the appropriate APF guidance formulation, depending on whether rest-to-rest or natural motion circumnavigation (NMC) maneuvers are being performed. If it is determined that a thrust maneuver is required, the appropriate attitude is computed and the spacecraft slews to this attitude in order to fire the thruster. Once complete, the LKP of the RSO is updated via the Clohessy-Wiltshire (CW) equations of relative motion, the anticipated attitude of the RSO is computed, and the spacecraft is commanded to slew to this attitude before repeating the process.

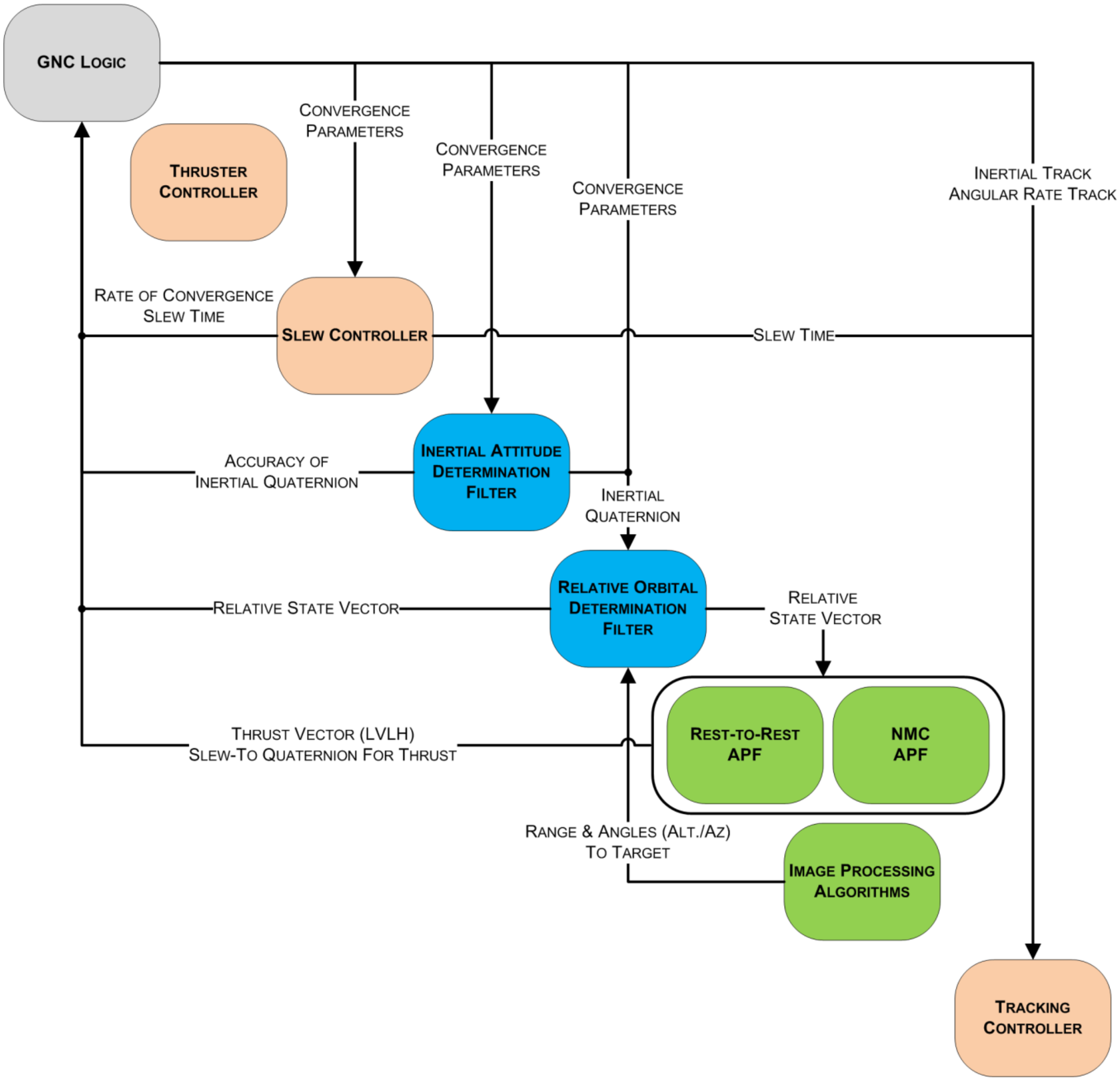


Figure 1. Top-level Prox-1 GN&C integration diagram.

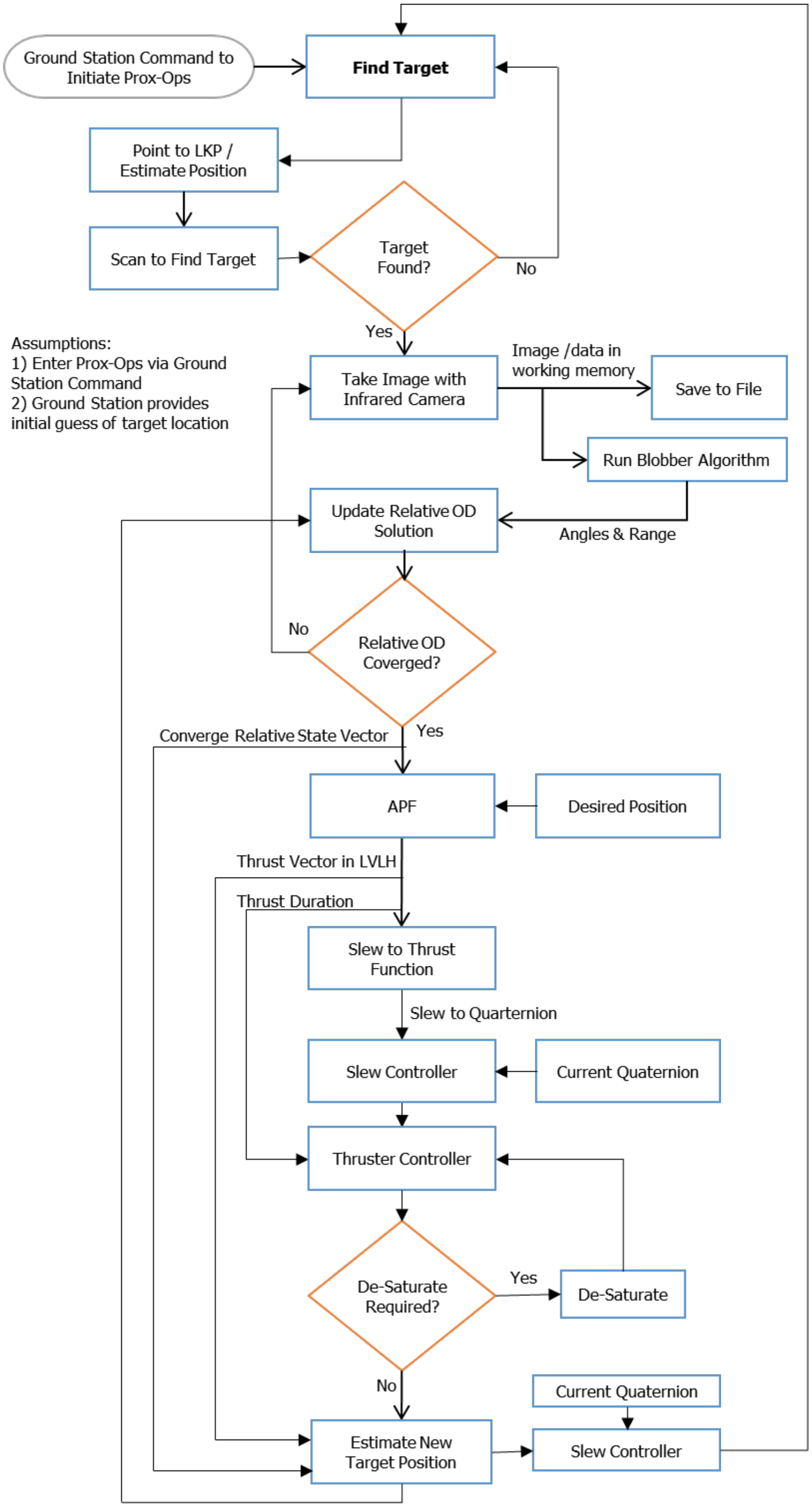


Figure 2. Generic proximity operations flow diagram.

2. Background Information

2.1. *Quaternion Primer*

2.1.1. Overview

A quaternion – also known as the Euler Parameters – is a sub set of hyper-complex numbers with rank 4. They were invented by William Rowan Hamilton in 1843, along with the following rule that governs operations on the vector part of the quaternion,

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{i} \otimes \boldsymbol{j} \otimes \boldsymbol{k} = -1 \tag{1}$$

where $\{\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$ are the dextral orthonormal basis in $\mathbb{R}^3$ and $\otimes$ represents the quaternion product, which will be discussed further [8].

The quaternion is a 4-tuple set $\boldsymbol{q} \epsilon \mathbb{R}^4$ that is composed of a vector element, $\boldsymbol{\varepsilon} \epsilon \mathbb{R}^3$ and a scalar component $\eta \epsilon \mathbb{R}$ such that

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{\varepsilon} \\ \eta \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \eta \end{bmatrix} \tag{2}$$

and has a unity norm constraint such that $\boldsymbol{q}^T \boldsymbol{q} = \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} + \eta^2 = 1$.

2.1.2. Equality and Additive Properties

In order for any two (2) quaternions to be equal, all the components must be same. That is, the two (2) quaternions $\boldsymbol{p}$

$$\boldsymbol{p} = \eta_p + \boldsymbol{i}p_1 + \boldsymbol{j}p_2 + \boldsymbol{k}p_3 \tag{3}$$

and $\boldsymbol{q}$

$$\boldsymbol{q} = \eta_q + \boldsymbol{i}q_1 + \boldsymbol{j}q_2 + \boldsymbol{k}q_3 \tag{4}$$

are equal if and only if $\eta_p = \eta_q, p_1 = q_1, p_2 = q_2, p_3 = q_3$.

The sum of the two (2) quaternions $\boldsymbol{p}$ and $\boldsymbol{q}$ listed above is defined by adding similar components such that

$$\boldsymbol{p} + \boldsymbol{q} = (\eta_p + \eta_q) + \boldsymbol{i}(p_1 + q_1) + \boldsymbol{j}(p_2 + q_2) + \boldsymbol{k}(p_3 + q_3) \tag{5}$$

Furthermore, each quaternion has an additive inverse, $-\boldsymbol{q}$, where each component is the negative of the corresponding component of $\boldsymbol{q}$.

2.1.3. Quaternion Multiplication

Similar to vectors in $\mathbb{R}^3$, the product of a scalar and a quaternion is simply some scalar, α, multiplied by all of the components of $\boldsymbol{q}$

$$\alpha\boldsymbol{q} = \alpha\eta_q + \boldsymbol{i}\alpha q_1 + \boldsymbol{j}\alpha q_2 + \boldsymbol{k}\alpha q_3 \tag{6}$$

The multiplication of any two (2) quaternions must satisfy the following products

$$\begin{aligned} &\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{i} \otimes \boldsymbol{j} \otimes \boldsymbol{k} = -1 && (a)\\ &\boldsymbol{ij} = \boldsymbol{k} = -\boldsymbol{ji} && (b)\\ &\boldsymbol{jk} = \boldsymbol{i} = -\boldsymbol{kj} && (c)\\ &\boldsymbol{ki} = \boldsymbol{j} = -\boldsymbol{ik} && (d) \end{aligned} \tag{7}$$

Using the Hamilton's special products listed and the rules for algebraic multiplication, the quaternion product follows. The product of two (2) quaternions $\boldsymbol{p}$ and $\boldsymbol{q}$ (listed above) is

$$\begin{aligned} \boldsymbol{pq} &= \left(\eta_p + \boldsymbol{i}p_1 + \boldsymbol{j}p_2 + \boldsymbol{k}p_3\right)\left(\eta_q + \boldsymbol{i}q_1 + \boldsymbol{j}q_2 + \boldsymbol{k}q_3\right) \\ &= \eta_p\eta_q + \boldsymbol{i}\left(p_1\eta_q + q_1\eta_p\right) + \boldsymbol{j}\left(p_2\eta_q + q_2\eta_p\right) + \boldsymbol{k}\left(p_3\eta_q + q_3\eta_p\right) \\ &\quad + \boldsymbol{i}^2 p_1 q_1 + \boldsymbol{ij} p_2 q_1 + \boldsymbol{ik}\, p_3 q_1 + \boldsymbol{ji} p_1 q_2 + \boldsymbol{j}^2 p_2 q_2 + \boldsymbol{jk} p_3 q_2 \\ &\quad + \boldsymbol{ki} p_1 q_3 + \boldsymbol{kj} p_2 q_3 + \boldsymbol{k}^2 p_3 q_3 \end{aligned} \tag{8}$$

Applying the special products and collecting like terms, the quaternion product becomes

$$\begin{aligned} \boldsymbol{pq} = {} & \eta_p\eta_q - (p_1q_1 + p_2q_2 + p_3q_3) \\ & + \eta_p(\boldsymbol{i}q_1 + \boldsymbol{j}q_2 + \boldsymbol{k}q_3) + \eta_q(\boldsymbol{i}p_1 + \boldsymbol{j}p_2 + \boldsymbol{k}p_3) \\ & + \boldsymbol{i}(p_2q_3 - p_3q_2) + \boldsymbol{j}(p_3q_1 - p_1q_3) + \boldsymbol{k}(p_1q_2 - p_2q_1) \end{aligned} \tag{9}$$

This can once more be rewritten as a more concise expression as

$$\boldsymbol{pq} = \eta_p\eta_q - \boldsymbol{p}\cdot\boldsymbol{q} + \eta_p\boldsymbol{q} + \eta_q\boldsymbol{p} + \boldsymbol{p}^x\boldsymbol{q} \tag{10}$$

where $\boldsymbol{p}^x$ is the skew operator defined as

$$\boldsymbol{a}^x = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{11}$$

and represents the matrix multiplication of a vector cross product.

2.2. *Quaternion Attitude Kinematics*

The attitude kinematics of quaternions are given as

$$\dot{\boldsymbol{q}} = \frac{1}{2}\underline{\Xi}(\boldsymbol{q})\boldsymbol{\omega} \tag{12}$$

where $\underline{\Xi}$: $\mathbb{R}^4 \rightarrow \mathbb{R}^{4x3}$ and is defined as

$$\underline{\Xi}(\boldsymbol{q}) = \begin{bmatrix} \eta\boldsymbol{I} + \boldsymbol{\varepsilon}^x \\ -\boldsymbol{\varepsilon}^T \end{bmatrix} \tag{1}$$

where $I \epsilon \mathbb{R}^{3x3}$ is the identity matrix.

3. Coordinate Frame Definitions & Transformations

3.1. *Overview*

This section will define the various the coordinate systems specific to the Prox-1 mission. Additionally, all relevant and pertinent coordinate system transformations will be derived as well.

3.2. *Coordinate Frame Definitions*

3.2.1. Imager Coordinate Frame (ICF) Definition

The main purpose of this coordinate frame is to provide relative information of the RSO – specifically angle and apparent size information. This will in turn be used to generate a relative position estimate between the chaser spacecraft and the RSO. The imager coordinate frame is defined with respect to the microbolometer's focal plane array (FPA). The coordinate frame is anchored to the top-left corner of the FPA when viewing the output image as shown in Figure 3.

The ICF dextral orthnormal basis vectors $\{\hat{\boldsymbol{X}}_I, \hat{\boldsymbol{Y}}_I, \hat{\boldsymbol{Z}}_I\}$are defined as follows:

- Origin of the ICF anchored in the upper-left hand corner of the FPA when viewing the output image.
- Imager (positive) X-axis: down the 480 pixel height of the FPA
- Imager (positive) Y-axis: to the right along the 640 pixel width of the FPA
- Imager (positive) Z-axis: via Right-Hand Rule (RHR)

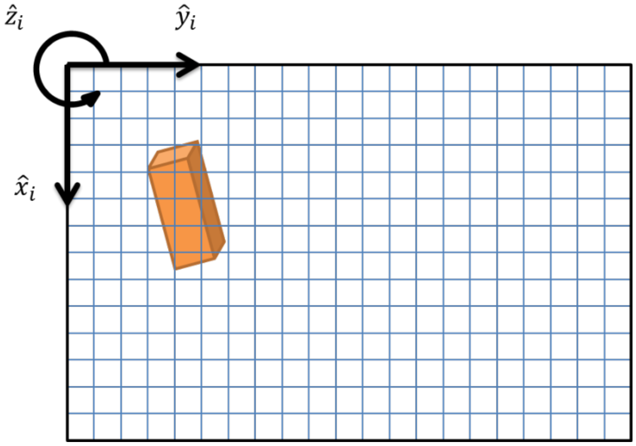


Figure 3. Imager coordinate frame

3.2.1. Earth-Centered Inertial (ECI) Coordinate Frame Definition

The ECI frame, also known as the Earth Centered Space Frame (ECSF), is a geocentric coordinate frame where the Earth's center is considered the origin. The basis vectors for the ECI frame are defined by the Conventional International Origin (CIO) reference axis of the North Pole's average location over the year 1900 and the direction of the vernal equinox (♈) at a given epoch. It is interesting to note that this coordinate frame is only quasi-inertial due to the motion of the Earth's center accelerating as it moves around the sun, however, it can be used as inertial here since no rotation takes place with respect to the CIO. A common formalization of the ECI frame for spaceflight applications is the J2000 frame. For this particular formalization this means that the coordinate frame is always centered at the center of Earth, the $\hat{Z}_{J2000}$ axis points toward the North Pole, the $\hat{X}_{J2000}$ direction points in the direction of the mean equinox at the epoch of 12:00 Terrestrial Time on January 1, 2000, and the $\hat{Y}_{J2000}$ direction is defined by the cross product of the first two basis vectors using the right hand rule. Note that this frame does not rotate with the spin of the Earth. Figure 1 shows a representation of the ECI Frame using the J2000 epoch [9].

3.2.1. Earth Centered, Earth Fixed (ECEF) Coordinate Frame Definition

The ECEF frame is similar to the ECI frame in that it is a geocentric frame, however, unlike the ECI frame, its basis vectors are fixed with respect to the Earth (i.e. the frame rotates with the spin of the Earth). The $\hat{Z}_{ECEF}$ axis points toward the CIO coinciding with the mean rotational axis of the Earth, not the instantaneous Earth rotational axis. The $\hat{X}_{ECEF}$ axis points toward the intersection of the equator and the prime meridian (the point of 0° latitude, 0° longitude) and the $\hat{Y}_{ECEF}$ is defined by the cross product of the first two basis vectors completing the right-handed system. Earth's magnetic field is one of many things conveniently expressed in the ECEF frame. Figure 4 shows the similarities and differences of the ECEF frame when compared to the ECI frame.

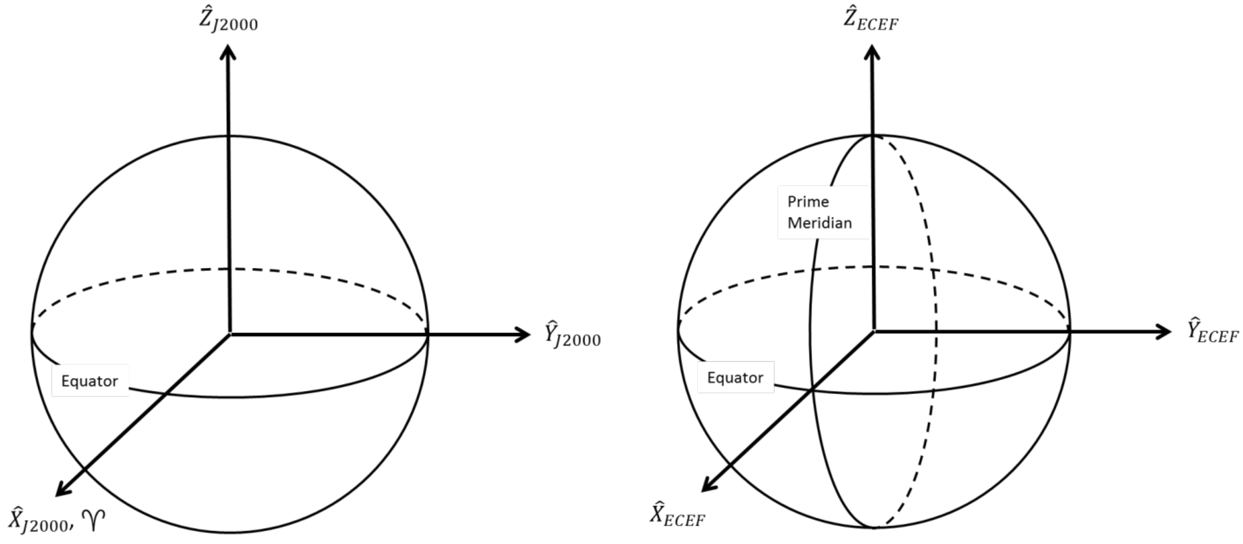


Figure 4. Comparison of the ECI and ECEF coordinate frame [9, 10]s.

3.2.2. Body-Fixed Coordinate Frame (BFCF) Definition

The main purpose of the Body-Fixed coordinate frame (BFCF) is to define the orientation of the attitude and control hardware. Specific to the Prox-1 spacecraft, the BFCF is illustrated in Figure 5. The BFCF dextral orthonormal basis vectors $\{\hat{\boldsymbol{X}}_B, \hat{\boldsymbol{Y}}_B, \hat{\boldsymbol{Z}}_B\}$ for the spacecraft are defined as follows:

- Origin of the BFCF anchored at the geometric center of the LVI plate.
- Spacecraft (positive) Y-axis normal to and thru the imager FPA
- Spacecraft (positive) Z-axis normal to and away (downward)
- Spacecraft (positive) X-axis via RHR

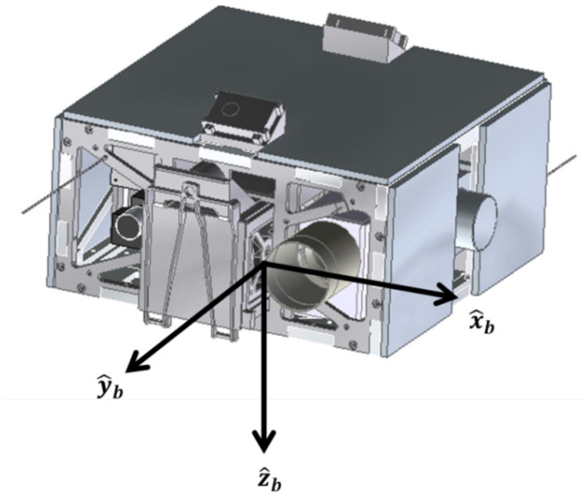


Figure 5. Prox-1dextral orthonormal basis superimposed on the spacecraft.

It is worth noting that the BFCF is *not* anchored at the spacecraft center of mass. This fact will become important later as the all measured properties of the spacecraft – including the moments of inertia – are taken about this geometrically fixed point [11, 12].

3.2.3. Satellite Coordinate Frame (RSW Frame) Definition

The Satellite Coordinate Frame, or as it is more commonly known as the RSW frame, is applies to the study of relative motion of a chaser spacecraft about the resident space object, or RSO. Typically, this coordinate frame is inertially fixed at the center of mass of the RSO with the dextral orthonormal basis being defined by the position unit vector (Radial) and the velocity unit vector (In-Track) of the RSO and the third basis vector (Cross-Track) via RHR. This orientation is illustrated in Figure 6.

Specific to the case of Prox-1, since no *a priori* knowledge of the RSO is known, the orientation of the RSW frame is based off of the Prox-1 inertial position, implying the following assumptions: $\frac{r_{rso}}{r_{chaser}}$ is small and $\hat{\boldsymbol{V}}_{rso} \approx \hat{\boldsymbol{V}}_{chaser}$. As a result, the orientation of the dextral orthonormal basis vectors $\{\hat{\boldsymbol{R}}, \hat{\boldsymbol{S}}, \widehat{\boldsymbol{W}}\}$ are defined as follows:

- $\hat{\boldsymbol{R}} := \frac{\boldsymbol{r}_{chaser}}{||\boldsymbol{r}_{chaser}||}$
- $\widehat{\boldsymbol{W}} := \frac{\boldsymbol{r}^x_{chaser}\boldsymbol{V}_{chaser}}{\left\|\boldsymbol{r}^x_{chaser}\boldsymbol{V}_{chaser}\right\|}$
- $\hat{\boldsymbol{S}} := \frac{\widehat{\boldsymbol{W}}^x\hat{\boldsymbol{R}}}{\left\|\widehat{\boldsymbol{W}}^x\hat{\boldsymbol{R}}\right\|}$

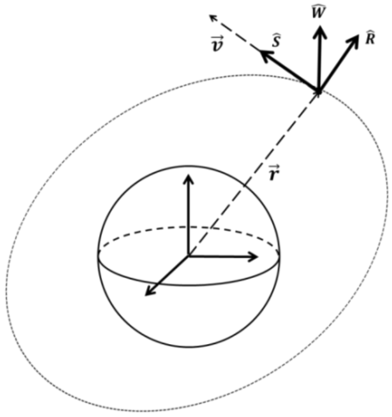


Figure 6. RSW orthonormal basis superimposed in the Earth-Centered Inertial (ECI) frame.

3.3. *Coordinate Frame Transformation*

3.3.1. Imager to Body-Fixed Coordinate Frame Transformation

Figure 3Figure 7 illustrates the imager coordinate frame (ICF) superimposed on the Prox-1 body-fixed coordinate frame (BFCF). Since these two (2) frames are spatially fixed the transformation can be derived visually. For the sake of simplicity, it is assumed that any mounting errors are small and negligible and therefore any angles between corresponding axes are approximately zero (0). The resulting coordinate frame transformation is

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_I \\ Y_I \\ Z_I \end{bmatrix} \qquad (2)$$

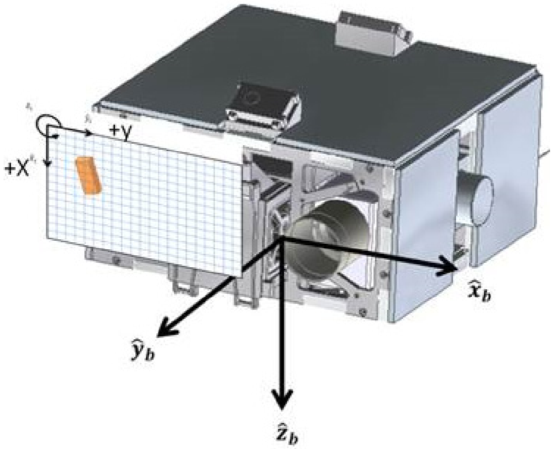


Figure 7. Imager coordinate frame superimposed on the Prox-1 body-fixed coordinate frame.

3.3.2. Body-Fixed to RSW Coordinate Frame Transformation

3.3.2.1. Coordinate Frame Transformation Derivation

The Body-Fixed to RSW coordinate frame transformation is essential as it allows for the determination of a relative state vector given measurements made in the BFCF. Figure 8 involved in the coordinate frame transformation formulation where: $\mathcal{J}$ is the inertial frame whose dextral orthonormal basis is defined by $\{\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}, \hat{\boldsymbol{Z}}\}$; $\mathcal{B}$ is the body-fixed coordinate frame whose dextral orthonormal basis is defined by $\{\hat{\boldsymbol{X}}_{\boldsymbol{B}}, \hat{\boldsymbol{Y}}_{\boldsymbol{B}}, \hat{\boldsymbol{Z}}_{\boldsymbol{B}}\}$ and orientation with respect to the inertial frame is defined by the quaternion $\boldsymbol{q}$; and $\mathcal{D}$ is the satellite coordinate frame, or RSW frame, whose dextral orthonormal basis is defined by $\{\hat{\boldsymbol{R}}, \hat{\boldsymbol{S}}, \hat{\boldsymbol{W}}\}$ and orientation quaternion with respect to the inertial frame is defined by the quaternion $\boldsymbol{q}_d$. Thus, the orientation of the BFCF with respect to the RSW frame is defined by the (error) quaternion $\boldsymbol{q}_e$.

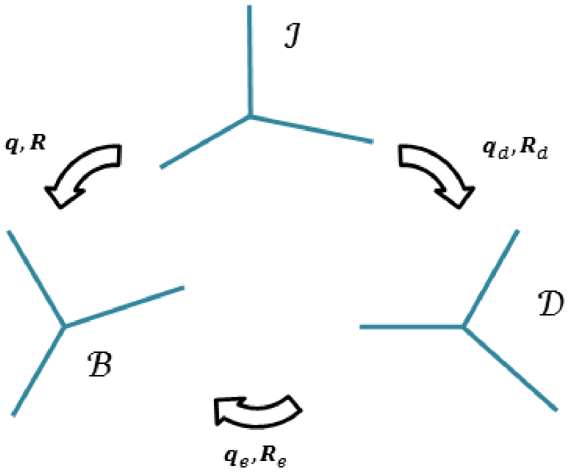

$\mathcal{J}$
$\boldsymbol{q}, \boldsymbol{R}$
$\boldsymbol{q}_d, \boldsymbol{R}_d$
$\mathcal{B}$
$\mathcal{D}$
$\boldsymbol{q}_e, \boldsymbol{R}_e$

Figure 8. Diagram of the relationship between the three (3) different reference frames.

Since the rotation matrix, $\underline{\boldsymbol{R}}_e$, which maps a vector in the RSW frame to the BFCF, is not known, a combination of the inertial quaternion and the definition of the RSW frame can be used to aid this transformation.

The rotation matrix that maps a vector in the inertial frame to a vector in the RSW frame can be defined as

$$\underline{\boldsymbol{R}}_d = [\widehat{\boldsymbol{R}} \quad \widehat{\boldsymbol{S}} \quad \widehat{\boldsymbol{W}}]^T \tag{3}$$

where $\widehat{\boldsymbol{R}} := \frac{r_{chaser}}{||\boldsymbol{r}_{chaser}||}$, $\widehat{\boldsymbol{W}} := \frac{r^{x}_{chaser}\boldsymbol{V}_{chaser}}{\left||r^{x}_{chaser}V_{chaser}\right||}$, and $\widehat{\boldsymbol{S}} := \frac{\widehat{\boldsymbol{W}}^{x}\widehat{\boldsymbol{R}}}{\left||\widehat{\boldsymbol{W}}^{x}\widehat{\boldsymbol{R}}\right||}$ and are column vectors anchored in the inertial (ECI) frame. Given a quaternion relating any two (2) frames, a rotation matrix can be formed that maps a vector between the frame can also be described as mapping between the orthonormal basis and the associated quaternion where $\underline{\boldsymbol{R}}_d\colon \mathbb{R}^3 \to \mathbb{R}^4$. Generically, the quaternion relating the orientation of the RSW frame with respect to the inertial frame can be given by the following set of equations, assuming $\eta_4 \neq 0$:two frames; resultantly, the rotation matrix transforming a vector from the inertial frame to the RSW

$$\begin{aligned} \eta_4 &= \pm\frac{1}{2}\sqrt{Tr(\underline{\boldsymbol{R}}) + 1} && (a)\\ \eta_1 &= \frac{1}{4\eta_4}\left(\underline{\boldsymbol{R}}_{2,3} - \underline{\boldsymbol{R}}_{3,2}\right) && (b)\\ \eta_2 &= \frac{1}{4\eta_4}\left(\underline{\boldsymbol{R}}_{3,1} - \underline{\boldsymbol{R}}_{1,3}\right) && (c)\\ \eta_3 &= \frac{1}{4\eta_4}\left(\underline{\boldsymbol{R}}_{1,2} - \underline{\boldsymbol{R}}_{2,1}\right) && (d) \end{aligned} \tag{4}$$

where $\underline{\boldsymbol{R}}(i,j)\epsilon\ \mathbb{R}$ for i,j = 1, 2, 3 and represents the (i,j) component of the rotation matrix. It is important to note that if η_4 is close to zero (0), Equations (4)(b-d) will not be accurate due to a loss of numerical precision in the square root. To maximize numerical accuracy, the set Equations (4), (5), (6), or (7) should be selected and evaluated based upon the largest argument in the square root [13].

$$\begin{aligned} \eta_1 &= \pm\frac{1}{2}\sqrt{1 + \underline{\boldsymbol{R}}_{1,1} - \underline{\boldsymbol{R}}_{2,2} - \underline{\boldsymbol{R}}_{3,3}} && (a)\\ \eta_2 &= \frac{1}{4\eta_4}\left(\underline{\boldsymbol{R}}_{1,2} + \underline{\boldsymbol{R}}_{2,1}\right) && (b)\\ \eta_3 &= \frac{1}{4\eta_4}\left(\underline{\boldsymbol{R}}_{1,3} + \underline{\boldsymbol{R}}_{3,1}\right) && (c)\\ \eta_4 &= \frac{1}{4\eta_4}\left(\underline{\boldsymbol{R}}_{2,3} - \underline{\boldsymbol{R}}_{3,2}\right) && (d) \end{aligned} \tag{5}$$

$$\begin{aligned}
\eta_2 &= \pm\frac{1}{2}\sqrt{1-\underline{\boldsymbol{R}}_{1,1}+\underline{\boldsymbol{R}}_{2,2}-\underline{\boldsymbol{R}}_{3,3}} && (a)\\
\eta_1 &= \frac{1}{4\eta_4}\left(\underline{\boldsymbol{R}}_{1,2}+\underline{\boldsymbol{R}}_{2,1}\right) && (b)\\
\eta_3 &= \frac{1}{4\eta_4}\left(\underline{\boldsymbol{R}}_{2,3}+\underline{\boldsymbol{R}}_{3,2}\right) && (c)\\
\eta_4 &= \frac{1}{4\eta_4}\left(\underline{\boldsymbol{R}}_{3,1}-\underline{\boldsymbol{R}}_{1,3}\right) && (d)
\end{aligned} \tag{6}$$

$$\begin{aligned}
\eta_3 &= \pm\frac{1}{2}\sqrt{1-\underline{\boldsymbol{R}}_{1,1}-\underline{\boldsymbol{R}}_{2,2}+\underline{\boldsymbol{R}}_{3,3}} && (a)\\
\eta_1 &= \frac{1}{4\eta_4}\left(\underline{\boldsymbol{R}}_{1,3}+\underline{\boldsymbol{R}}_{3,1}\right) && (b)\\
\eta_2 &= \frac{1}{4\eta_4}\left(\underline{\boldsymbol{R}}_{2,3}+\underline{\boldsymbol{R}}_{3,2}\right) && (c)\\
\eta_4 &= \frac{1}{4\eta_4}\left(\underline{\boldsymbol{R}}_{1,2}-\underline{\boldsymbol{R}}_{2,1}\right) && (d)
\end{aligned} \tag{7}$$

Now that the quaternion relating the orientation of the RSW frame with respect to the inertial frame, $\boldsymbol{q}_d$, is known, the error quaternion relating the orientation of the BFCF with respect to the RSW frame can be computed. Without the loss of generality, the error quaternion, $\boldsymbol{q}_e$, is defined as

$$\boldsymbol{q}_e = \boldsymbol{q}_\mathbf{1} \otimes \boldsymbol{q}_\mathbf{2} \tag{8}$$

When computed, the error quaternion simplifies to

$$\boldsymbol{q}_e = \left[\,\underline{\Xi}\left(\boldsymbol{q}_\mathbf{2}^{-\mathbf{1}}\right)\boldsymbol{q}_2^{-1}\,\right]\boldsymbol{q}_1 \tag{9}$$

Specific to the application of orientations of the RSW and BFCF frames, the error quaternion is

$$\boldsymbol{q}_e = \left[\,\underline{\Xi}\left(\boldsymbol{q}_d^{-1}\right)\boldsymbol{q}_d^{-1}\,\right]\boldsymbol{q} \tag{10}$$

Before continuing, it is worth spending a few lines to show the derivation of a quaternion-based rotation operator. Generally speaking, to rotate a vector from one frame to another, the following transformation is used

$$\boldsymbol{w} = \underline{\boldsymbol{A}}\boldsymbol{v} \tag{11}$$

where a vector $\boldsymbol{v}, \boldsymbol{v} \,\boldsymbol{\epsilon}\, \mathbb{R}^\mathbf{3}$ is operated on by a rotation matrix $\underline{\boldsymbol{A}}, \underline{\boldsymbol{A}} \,\epsilon\, \mathbb{R}^{3x3}$, and whose output in the desired frame is $\boldsymbol{w}, \boldsymbol{w} \,\boldsymbol{\epsilon}\, \mathbb{R}^\mathbf{3}$. In order to use the quaternion to rotate a vector from to one frame to another, both vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ must first be transformed into pure quaternions to allow the use of a quaternion as the rotation operator – that is treat a vector $\boldsymbol{v} \,\epsilon\, \mathbb{R}^3$ as if it were a quaternion

$\boldsymbol{q}\ \epsilon\ \mathbb{R}^4$ whose real part is zero (0). Therefore, the vector $\boldsymbol{v}$ becomes $\begin{bmatrix}\boldsymbol{v}\\0\end{bmatrix}$; likewise, the vector $\boldsymbol{w}$ becomes $\begin{bmatrix}\boldsymbol{w}\\0\end{bmatrix}$. Since both $\boldsymbol{v}$ and w are now in $\mathbb{R}^4$, the rotation operator can now be represented by a quaternion. Equation (11) becomes

$$\boldsymbol{w} = \boldsymbol{q} \otimes \boldsymbol{v} \otimes \boldsymbol{q}^{-1} \tag{12}$$

which can be re-written as a matrix equation

$$\boldsymbol{w} = \left((\eta^2 - \boldsymbol{\varepsilon}^{\boldsymbol{T}}\boldsymbol{\varepsilon})\underline{\boldsymbol{I}} + \boldsymbol{2}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^{\boldsymbol{T}} - \eta\boldsymbol{\varepsilon}^{\boldsymbol{x}})\right)\boldsymbol{v} \tag{13}$$

Evaluating the matrix equation defined by Equation (13) yields the rotation matrix as a function of a quaternion

$$\boldsymbol{w} = \begin{bmatrix} 2\eta^2 - 1 + 2q_1^2 & 2q_1q_2 - 2\eta q_3 & 2q_1q_3 + 2\eta q_2 \\ 2q_1q_2 + 2\eta q_3 & 2\eta^2 - 1 + 2q_2^2 & 2q_2q_3 - 2\eta q_1 \\ 2q_1q_3 - 2\eta q_2 & 2q_2q_3 + 2\eta q_1 & 2\eta^2 - 1 + 2q_3^2 \end{bmatrix}\boldsymbol{v} \tag{14}$$

Defining the quaternion-based rotation matrix as $\underline{\boldsymbol{Q}}(\boldsymbol{q})$, the transformation can be written in a more familiar form,

$$\boldsymbol{w} = \underline{\boldsymbol{Q}}(\boldsymbol{q})\boldsymbol{v} \tag{15}$$

where $\underline{\boldsymbol{Q}}, \underline{\boldsymbol{Q}}\ \epsilon\ \mathbb{R}^{3x3}$, is the rotation operator given a quaternion $\boldsymbol{q}, \boldsymbol{q}\ \epsilon\ \mathbb{R}^4$. From this, it can be seen that the vector component of the quaternion describes the axis of rotation.

Continuing, since the error quaternion and the corresponding rotation matrix is known, the transformation from the BFCF to the RSW frame is given as

$$\begin{bmatrix} R \\ S \\ W \end{bmatrix} = \underline{\boldsymbol{Q}}^T(\boldsymbol{q}_{\boldsymbol{e}})\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \tag{16}$$

Given the definition of the inverse of a quaternion, Equation (16) can be rewritten as

$$\begin{bmatrix} R \\ S \\ W \end{bmatrix} = \underline{\boldsymbol{Q}}(\boldsymbol{q}_{\boldsymbol{e}}^{-1})\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \tag{17}$$

where

$$\boldsymbol{q}_e^{-1} = \begin{bmatrix} -\boldsymbol{\varepsilon}_e \\ \eta_e \end{bmatrix} \tag{18}$$

3.3.2.2. BFCF to RSW Frame Algorithm

Given the computational and system resource limitations of many (space-based) in-situ computational platforms, it is highly beneficial to utilize low-cost computational algorithms with a minimal memory footprint. To accomplish this, computationally costly operations such as matrix inversion and square root function calls, for example, are to be minimized. To aid with minimizing the memory footprint of an algorithm, selecting the appropriate data type for the level of precision is critical. For example, if the resulting inertial quaternion from a navigation filter only produces four (4) digits places of precision, then the smallest data type that should be used are 'Single' floating point values; conversely, if a navigation filter is capable of seven (7) significant figures, then at least a 'Double' floating point value should be utilized as it is capable of up to 16 digits of precision. For higher digits of precision, a 'Long Double' floating data type should be used as it is capable of up to 34 digits of precision [14-16]. The resulting algorithm is listed in Table 2.

The input and output variables for the BFCF to RSW Transformation algorithm are listed in Table 1. At the time of writing, the precision of the resulting inertial quaternion from the navigation filter is unknown. Single' floating point data types can also be used to store each value of the inertial position and velocity

Table 3 illustrates the algorithm overhead measured in terms of variable storage footprint. For the purpose of this analysis, it is assumed all inputs are passed by value. One method to reduce the overall memory footprint of the algorithm is to pass the inputs by address rather than value. Passing a variable by address (via a pointer in C/C++), does not replicate the variable and results in lower memory usage. The associated risks by passing an input via pointer include modification of the value of input at a particular memory address, which will impact future computations involving that variable until it is updated. However, these concerns are not addressed here as 'Best Coding Practices' are outside the scope of this section and are mentioned only for thought-provoking reasons.

Table 1. BFCF to RSW Frame Algorithm Inputs and Outputs.

Variable Definition		*Data Type*
Inputs	Inertial Quaternion, $\boldsymbol{q}$: 4x1 Vector	Single
	Inertial Position, $\boldsymbol{r}_{chaser}$: 3x1 Vector	Single
	Inertial Velocity, $\boldsymbol{V}_{chaser}$: 3x1 Vector	Single
	Body-Fixed Measurement, $\boldsymbol{X}_{B}$: 3x1 Vector	Single
Output	RSW Measurement, $\boldsymbol{X}_{RSW}$: 3x1 Vector	Single

Table 2. BFF to RSW Coordinate Frame Transformation Algorithm

Body-Fixed to RSW Coordinate Frame Transformation Algorithm & Variable Definition	
1. Compute the components of the rotation matrix $\underline{\boldsymbol{R}}_d = [\widehat{\boldsymbol{R}} \quad \widehat{\boldsymbol{S}} \quad \widehat{\boldsymbol{W}}]^T$ where $\widehat{\boldsymbol{R}} := \frac{\boldsymbol{r}_{chaser}}{\lVert\boldsymbol{r}_{chaser}\rVert}, \widehat{\boldsymbol{W}} := \frac{\boldsymbol{r}^x_{chaser}\boldsymbol{V}_{chaser}}{\lVert\boldsymbol{r}^x_{chaser}\boldsymbol{V}_{chaser}\rVert}$, and $\widehat{\boldsymbol{S}} := \frac{\widehat{\boldsymbol{W}}^x\widehat{\boldsymbol{R}}}{\lVert\widehat{\boldsymbol{W}}^x\widehat{\boldsymbol{R}}\rVert}$	Variable Definition: $\underline{\boldsymbol{R}}_d$: 3x3 Matrix Data Type: Single
2. Evaluate only the arguments of Equations (4)a, (5)a, (6)a, and (7)a: Equation (4)a: $\eta_4 = \pm\frac{1}{2}\sqrt{Tr(\underline{\boldsymbol{R}}) + 1}$ Equation (5)a: $\eta_1 = \pm\frac{1}{2}\sqrt{1 + \underline{\boldsymbol{R}}_{1,1} - \underline{\boldsymbol{R}}_{2,2} - \underline{\boldsymbol{R}}_{3,3}}$ Equation (6)a: $\eta_2 = \pm\frac{1}{2}\sqrt{1 - \underline{\boldsymbol{R}}_{1,1} + \underline{\boldsymbol{R}}_{2,2} - \underline{\boldsymbol{R}}_{3,3}}$ Equation (7)a: $\eta_3 = \pm\frac{1}{2}\sqrt{1 - \underline{\boldsymbol{R}}_{1,1} - \underline{\boldsymbol{R}}_{2,2} + \underline{\boldsymbol{R}}_{3,3}}$	Variable Definition: η: Scalar Date Type: Single
3. Determine the largest value between $\eta_1, \eta_2, \eta_3, \eta_4$ if ($\eta_1 > \{\eta_2, \eta_3, \eta_4\}$) • Use Equations (5) else if ($\eta_2 > \{\eta_1, \eta_3, \eta_4\}$) • Use Equations (6) else if ($\eta_3 > \{\eta_2, \eta_1, \eta_4\}$) • Use Equations (7) else: • Use Equations (4)	Variable Definition: **N/A** Data Type: **N/A**
4. Compute $\boldsymbol{q}_d$ using the appropriate set of equations determined from Step 3.	Variable Definition: $\boldsymbol{q}_d$: 4x1 Vector Data Type: Single
5. Determine the error quaternion $\boldsymbol{q}_e$ $q_e = [\,\boldsymbol{\Xi}(\boldsymbol{q}_d^{-1})\ \boldsymbol{q}_d^{-1}\,]\boldsymbol{q}$	Variable Definition: $\boldsymbol{q}_e$: 4x1 Vector Data Type: Single

Body-Fixed to RSW Coordinate Frame Transformation Algorithm & Variable Definition	
6. Compute the rotation operator $\underline{\boldsymbol{Q}}(\boldsymbol{q}_{\boldsymbol{e}}^{-1}) = \underline{\boldsymbol{Q}}\left(\begin{bmatrix}-\boldsymbol{\varepsilon}_e\\ \eta_e\end{bmatrix}\right)$ $\underline{\boldsymbol{Q}}\left(\begin{bmatrix}-\boldsymbol{\varepsilon}_e\\ \eta_e\end{bmatrix}\right) = (\eta_e^2 - \boldsymbol{\varepsilon}_e^T\boldsymbol{\varepsilon}_e)\underline{\boldsymbol{I}} + 2(\boldsymbol{\varepsilon}_e\boldsymbol{\varepsilon}_e^T + \eta_e\boldsymbol{\varepsilon}_e^x)$ *(Note: This equation differs from Equation (13) by a single sign difference as it takes into account that the rotation operator is being computed for an inverse quaternion as opposed to a quaternion.)*	Variable Definition: $\underline{\boldsymbol{Q}}$: 3x3 Matrix Data Type: Single
7. Transform vector from BFCF to RSW $\begin{bmatrix}R\\ S\\ W\end{bmatrix} = \underline{\boldsymbol{Q}}(\boldsymbol{q}_{\boldsymbol{e}}^{-1})\begin{bmatrix}X_B\\ Y_B\\ Z_B\end{bmatrix}$	Variable Definition: $\boldsymbol{X}_{RSW}$: 3x1 Vector Data Type: Single

Table 3. Approximate variable storage footprint for the BFCF to RSW transformation algorithm

Parameter	*Value*
Input Variable Storage Footprint	416 bits, 52 bytes
In-Function Variable Storage Footprint	1056 bits, 132 bytes
Total Variable Storage Footprint	1472 bits, 184 bytes

3.3.3. RSW to ECI Coordinate Frame Transformation

The rotation matrix that maps a vector in the inertial frame to a vector in the RSW frame can be defined as

$$\underline{\boldsymbol{R}}_d = [\hat{\boldsymbol{R}} \quad \hat{\boldsymbol{S}} \quad \widehat{\boldsymbol{W}}]^T \tag{19}$$

where $\hat{\boldsymbol{R}} := \frac{r_{chaser}}{||r_{chaser}||}$, $\widehat{\boldsymbol{W}} := \frac{r_{chaser}^x\boldsymbol{V}_{chaser}}{||r_{chaser}^x\boldsymbol{V}_{chaser}||}$, and $\hat{\boldsymbol{S}} := \frac{\widehat{\boldsymbol{W}}^x\hat{\boldsymbol{R}}}{||\widehat{\boldsymbol{W}}^x\hat{\boldsymbol{R}}||}$ and are column vectors anchored in the inertial (ECI) frame. Given the rotation matrix, the resulting transformation to map a vector from the RSW frame to the ECI frame is

$$\begin{bmatrix}X\\ Y\\ Z\end{bmatrix} = \underline{\boldsymbol{R}}_d^T\begin{bmatrix}R\\ S\\ W\end{bmatrix} \tag{20}$$

3.3.4. ECEF to ECI Coordinate Frame Transformation

In order to transform between any two frames, three rotational matrixes, $R_1, R_2,$ and R_3 must first be defined:

$$R_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}, \quad R_2(\phi) = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix},$$
$$R_3(\phi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

where ϕ can be used to represent any angle that the system is being transformed by. To complete the transformation from the ECI frame to the ECEF frame, four types of motion need to be accounted for: precession, nutation, rotation, and polar motion. If three types of these motions are neglected, precession, nutation, and polar motion, then the transformation between the two frames is relatively simple and only rotation needs to be accounted for. The rotational matrix is as follows:

$$\tilde{X}_{ECEF} = R_3(\theta_{GST})\tilde{X}_{ECI} \tag{14}$$

where θ_{GST} is the rotational angle based on the Greenwich Sidereal Time (GST).

In Prox-1's case, however, this type of simplification cannot be done. Since autonomous proximity operations will be taken place throughout the mission, the transformation into the ECEF frame needs to be more accurate. For that reason, transformation between the ECI and ECEF coordinate frames was performed with assistance for verification purposes of the constants[10]. To begin, a visualization of the precession and nutation phenomenon may help. Figure 3 shows the precession and nutation the earth undergoes over the long, 26,000 year precession period and the shorter, 18.6 year nutation period.

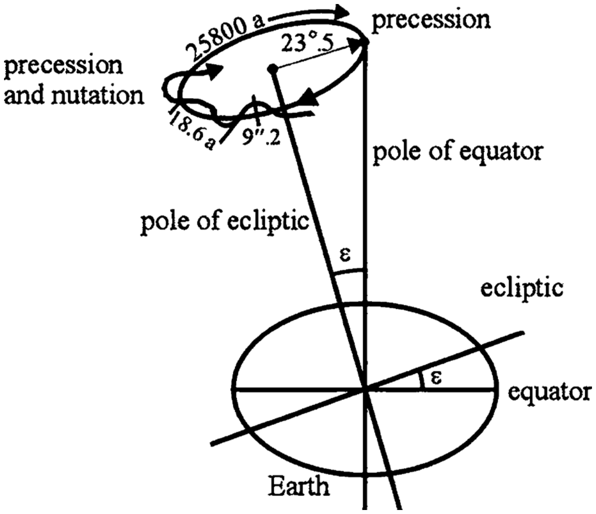


Figure 3: Precession and Nutation [10]

In order to transform the Earth Centered Inertial frame into the Earth Centered, Earth Fixed frame, a series of rotations needs to be completed in a specific order. The four rotations take place as followed:

$$\tilde{X}_{ECEF} = R_M R_S R_N R_P \tilde{X}_{ECI}$$

where R_M is polar motion matrix, R_S is the Earth rotation matrix, R_N is the nutation matrix, R_P is the precession matrix, and $\tilde{X}$ is the coordinate vector with subscripts denoting the related coordinate system.

3.3.4.1. Precession

The Earth's rotational axis is not fixed in space, but rather has a slow precession period that takes place over a 26,000 year period (about 1^{o} per 72 years). This precession is essentially a torque acting on the Earth by a combination of the sun's and moon's gravity pulling on the equatorial bulges on the Earth (note: other planets gravity pulls on the Earth as well, but the amount is insignificant compared to the sun and moon). The precession rotational matrix is defined as:

$$R_p = R_3(-z)R_2(\theta)R_3(-\zeta)$$

or

$$R_P = \begin{bmatrix} \cos(z)\cos(\theta)\cos(\zeta) - \sin(z)\sin(\zeta) & -\cos(z)\cos(\theta)\sin(\zeta) - \sin(z)\sin(\zeta) & -\cos(z)\sin(\theta) \\ \sin(z)\cos(\theta)\cos(\zeta) + \cos(z)\sin(\zeta) & -\sin(z)\cos(\theta)\sin(\zeta) + \cos(z)\cos(\zeta) & -\sin(z)\sin(\theta) \\ \sin(\theta)\cos(\zeta) & -\sin(\theta)\sin(\zeta) & \cos(\theta) \end{bmatrix}$$

where z, θ, ζ are precession parameters defined as:

$$\begin{gathered} z = 2306.^{\prime\prime}\,2181T + 1.^{\prime\prime}09468T^2 + 0.^{\prime\prime}\,018203T^3 \\ \theta = 2004.^{\prime\prime}\,3109T - 0.^{\prime\prime}42665T^2 - 0.^{\prime\prime}\,018203T^3 \\ \zeta = 2306.^{\prime\prime}\,2181T - 0.^{\prime\prime}30188T^2 - 0.^{\prime\prime}\,017998T^3 \end{gathered}$$

where T is the measuring time in Julian centuries (36525 days) counted from J2000, 0h.

3.3.4.2. Nutation

Nutation is principally caused by the tidal forces of the sun and moon. The largest component of nutation is caused by the moon's orbital nodes which occur with an 18.6 year period. Smaller components of nutation range in period from 14 days up to 18.6 years. The rotational matrix associated with nutation is:

$$R_N = R_1(-\varepsilon - \Delta\varepsilon)R_3(-\Delta\psi)R_1(\varepsilon)$$

or

$$R_N = \begin{bmatrix} \cos(\Delta\psi) & -\sin(\Delta\psi)\cos(\varepsilon) & -\sin(\Delta\psi)\sin(\varepsilon) \\ \sin(\Delta\psi)\cos(\varepsilon_t) & \cos(\Delta\psi)\cos(\varepsilon_t)\cos(\varepsilon) + \sin(\varepsilon_t)\sin(\varepsilon) & \cos(\Delta\psi)\cos(\varepsilon_t)\sin(\varepsilon) - \sin(\varepsilon_t)\cos(\varepsilon) \\ \sin(\Delta\psi)\sin(\varepsilon_t) & \cos(\Delta\psi)\sin(\varepsilon_t)\cos(\varepsilon) - \cos(\varepsilon_t)\sin(\varepsilon) & \cos(\Delta\psi)\sin(\varepsilon_t)\sin(\varepsilon) + \cos(\varepsilon_t)\cos(\varepsilon) \end{bmatrix}$$

which can be approximated by letting $\cos(\Delta\psi) = 1$ and $\sin(\Delta\psi) = \Delta\psi$ for very small angles of $\Delta\psi$, as is the case for the Prox-1 mission. Therefore,

$$R_N \approx \begin{bmatrix} 1 & -\Delta\psi\cos(\varepsilon) & -\Delta\psi\sin(\varepsilon) \\ \Delta\psi\cos(\varepsilon_t) & 1 & -\Delta\varepsilon \\ \Delta\psi\sin(\varepsilon_t) & \Delta\varepsilon & 1 \end{bmatrix}$$

where ε is the mean axial tile of the ecliptic angle on the date as seen in Figure 3, $\Delta\psi$ and $\Delta\varepsilon$ are nutation angles in longitude and axial tilt, and $\varepsilon_t = \varepsilon + \Delta\varepsilon$. ε is further defined as:

$$\varepsilon = 84381.''448 - 46''.8150T - 0''.00059T^2 + 0.''001813T^3$$

3.3.4.3. Earth rotation

Since the Earth rotation is only about the $\hat{Z}_{ECI}$-*axis* based on the Greenwich Apparent Sidereal Time (GAST), the Earth rotation matrix is simply represented as:

$$R_S = R_3(\text{GAST})$$

where GAST is equal to:

$$\text{GAST} = \text{GMST} + \Delta\psi\cos(\varepsilon) + 0.''00264\sin(\Omega) + 0.''000063\sin(2\Omega)$$

where GMST is the Greenwich Mean Sidereal Time, Ω is the mean longitude of the ascending node of the moon, and ε is as defined above. Additionally,

$$\text{GMST} = \text{GMST}_0 + \alpha\text{UT1}$$

$$\text{GMST}_0 = 6 * 3600.''0 + 41 * 60.''0 + 50.''54841 + 8640184.''812866T_0 + 0.''093104T_0^2 - 6.''2 * 10^{-6}T_0^3$$

$$\alpha = 1.002737909350795 + 5.9006 * 10^{-11}T_0 - 5.9 * 10^{-15}T_0^2$$

where GMST$_0$ is the Greenwich Mean Sidereal Time at midnight on the day of interest, α is the rate of change, UT1 is the polar motion corrected for Universal Time, and T_0 is the measuring time in Julian Centuries (36525 days) counted from J2000.0 to 0h UT1 on the measuring day.

3.3.4.4. Polar Motion

Polar motion accounts for the difference between the true equatorial system and the ECEF system. In other words, polar motion is simply the angle between the pole on the date measured and the CIO pole. Since polar motion is defined in the *xy-plane* associated with an *x-axis* pointed south corresponded to the mean Greenwich meridian and *y-axis* pointed to the west, two angles x_p and y_p can be defined as the angles of the pole on the date of measurement. This means the polar rotational matrix is defined as:

$$R_M = R_2(-x_p)R_1(-y_p) = \begin{bmatrix} \cos(x_p) & \sin(x_p)\sin(y_p) & \sin(x_p)\cos(y_p) \\ 0 & \cos(y_p) & -\sin(y_p) \\ -\sin(x_p) & \cos(x_p)\sin(y_p) & \cos(x_p)\cos(y_p) \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & x_p \\ 0 & 1 & -y_p \\ -x_p & y_p & 1 \end{bmatrix}$$

Since both the quantities of x_p and y_p are small angles, the small angle approximations for sine and cosine can be made without significant loss in results. Using the International Earth Rotation and Reference System Service (IERS) website, x_p and y_p can be looked up directly for the dates needed.

4. Guidance Formulation

4.1. *Image Processing Algorithms (IPAs)*

4.1.1. Imaging Instruments

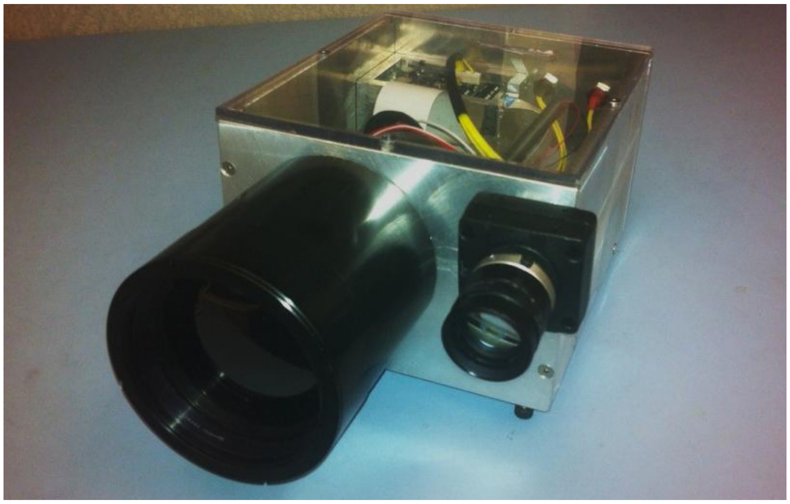


Figure 9. Visible camera (right) and MTI (left) mounted together.

The Prox-1 spacecraft will utilize two (2) independent imagers to locate RSOs: a visible camera and a Microbolometer Thermal Imager (MTI). The visible camera is a Point Grey Research CMLN-13SC Chameleon camera. This Commercial Off-The-Shelf (COTS) camera features a 1296-by-964 pixel (1.3MP) Red-Green-Blue (RGB) CCD and utilizes a Fujinon HF25HA-1B lens providing a FOV of 10.97º by 8.16º. The MTI features a Focal Plane Array (FPA) size of 640-by-480 and utilizes a 100mm lens that produces a FOV of 9.1º by 6.8º. It is important to note that the MTI will be the main imager since it will be able to track the RSO in all lighting conditions.

The desired frequency of imaging is to be determined by the requirements of the APF decision-making process. Two opposing parameters must be balanced in this requirement. First, the gradient-based search techniques of the APF process are more optimal when state information is available with high frequency. In this case the state information is provided by the IPAs following new image capture. Therefore, the higher frequency of image capture, the more likely it is that APF processes will close to a solution and the solution is likely to be more efficient for time and fuel. However, this desire is opposed by the limited computational abilities of Prox-1. Image capture, transfer, and image processing are computationally expensive and will tax the computing budget. A more detailed simulation must be utilized to further evaluate the requirement.

4.1.2. Image Processing Algorithms

4.1.2.1. Overview

The image processing algorithms (IPAs) form the basis of the Prox-1 guidance as they provide relative positioning information to the rest of the Prox-1 GN&C system. While in their current form, the IPAs results are shown to vary depending on both the range and orientation of the RSO. However, through the use of filtering, the precision of the relative position estimate is increased.

4.1.2.2. Blobber Algorithm

Immediately following image acquisition, the Blobber Algorithm is the first step in the IPAs and the main process in image processing. The purpose of the Blobber Algorithm is to identify the RSO in the image. The algorithm flow diagram is illustrated in Figure 10.

First, it detects the pixels in an image which are in a certain range of intensity; for the MTI this intensity is analogous to temperature; for the VC it is a measure of visible radiance. Next, it detects the groups of pixels which are connected to each other and which form a blob (Binary Large OBject). Generally, there is not just one blob detected, especially when the image background is the Earth. However the object which has to be localized should be one of the blobs detected. Therefore, to choose the correct blob among the few which are remaining, an area screening must be applied. The correct blob is selected by calculating its area on the image, actually the number of pixels contained in the blob. Indeed, with previous knowledge about the object to detect, it is possible to know approximately the range of the cross-sectional area [17].

The ideal results of the Blobber algorithm are a calculated area for the identified blob and the location of the Center of Brightness (COB) for the blob. The COB is similar to an area centroid of or a center of mass and is used as the central location of the RSO. The COB coordinates are given with respect to the imager's focal plane; therefore these coordinates must be mapped into the body-fixed coordinate frame in order to represent the unit vector from Prox-1 to the RSO.

4.1.2.1. Unit Vector Determination

Using the COB coordinates, a unit vector can be found using geometry and the optical properties of the imager. The RSO can at all times be considered to be focused at infinity in relation to the focal length of the camera lenses – that is the distance to the RSO is much greater than the focal length of the lens –the calculation of the unit vector or rotation angles can be determined.

As illustrated by Figure 3, the unit vector, $\vec{\boldsymbol{u}}$**,** can be determined given the focal length of the lens and the position of the COB on the FPA or CCD. Alternatively, the position vector of RSO can be expressed in spherical coordinates using the radial distance of the COB and the rotation angles θ_i and φ_i. Using the rotation defined in Equation **Error! Bookmark not defined.**, the unit vector describing the relative position of the RSO to Prox-1 is given as

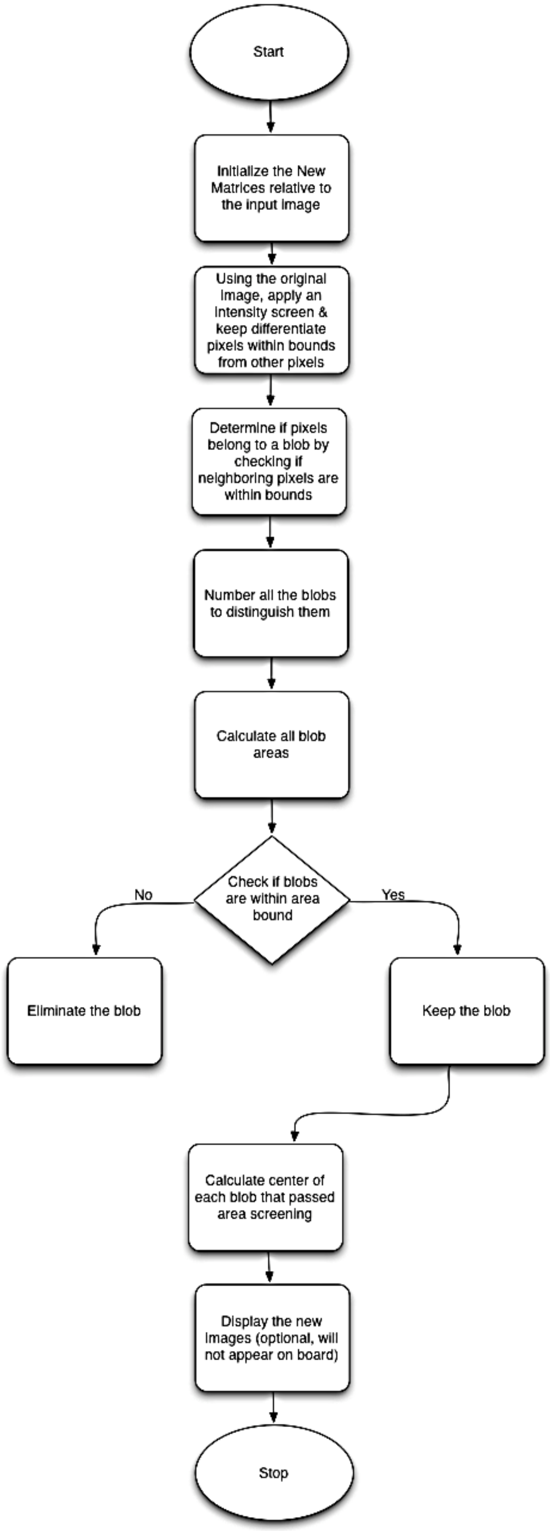

Start
Initialize the New Matrices relative to the input image
Using the original image, apply an intensity screen & keep differentiate pixels within bounds from other pixels
Determine if pixels belong to a blob by checking if neighboring pixels are within bounds
Number all the blobs to distinguish them
Calculate all blob areas
Check if blobs are within area bound
No
Yes
Eliminate the blob
Keep the blob
Calculate center of each blob that passed area screening
Display the new images (optional, will not appear on board)
Stop

Figure 10. Blobber algorithm flow diagram[18]

$$\begin{bmatrix} \hat{X}_B \\ \hat{Y}_B \\ \hat{Z}_B \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \sin(\theta_i)\cos(\varphi_i) \\ \sin(\theta_i)\sin(\varphi_i) \\ \cos(\theta_i) \end{bmatrix} \tag{15}$$

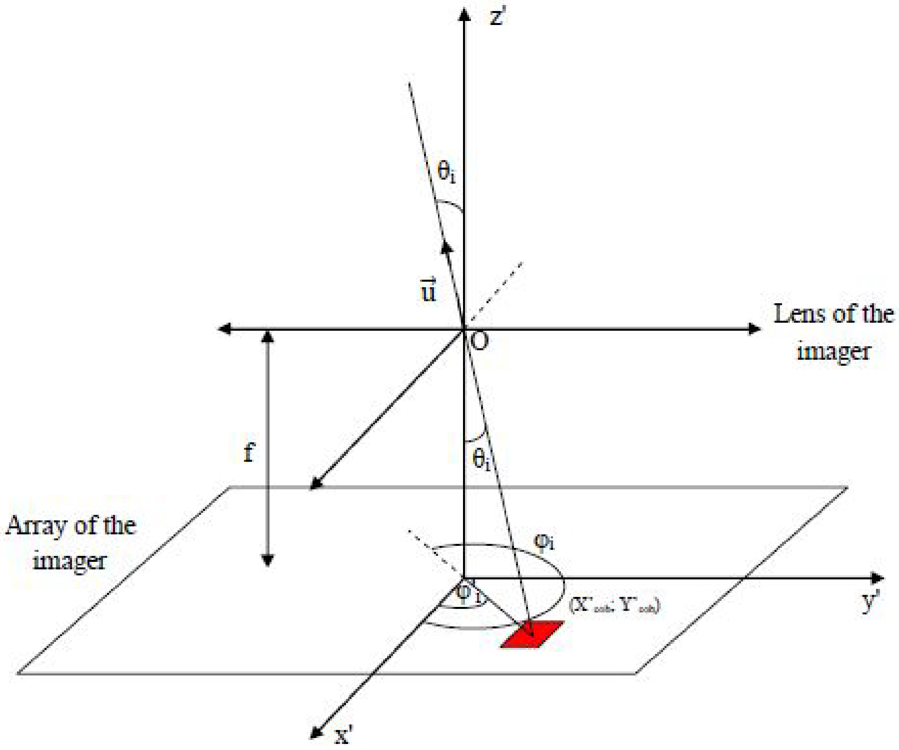


Figure 11. Geometric representation of the relative position vector in the ICF.

4.1.2.2. Range Estimation

Range estimation occurs only after the RSO has been successfully identified and the unit vector from the Prox-1 spacecraft to the RSO is determined. Range estimation can be broken down into four main steps:

1) Determination of the major and minor axes of the RSO
2) Determination of the ratio of the major axis to the minor axis
3) Estimation of the minimum and maximum apparent areas of the RSO.
4) Determination of the range its associated uncertainty

4.1.2.2.1. Major and Minor Axis Determination

The major axis of the RSO is the line which maximizes the sum of the squares of the distance between itself and each pixel of the blob belonging to the RSO. Likewise, the minor axis of the RSO is the line which minimizes the sum of the distance between itself and the blob. By definition, these two (2) axes intersect at the center of the blob, which is defined as the COB, and are orthogonal to each other.

4.1.2.2.2. Major-to-Minor Axis Ratio Determination

The ratio of the major-to-minor axis is determined by comparing the lengths of each axis. The length of each axis is determined by counting the number of pixels along the line defining the respective axes. This can be accomplished by rotating the image of the blob such that both the major axis is vertical and the minor axis is horizontal. Since the two (2) axes intersect at the COB, the column and row of the major and minor axis are known respectively. Furthermore, since the blob is a binary matrix – that is, the matrix is populated by ones (1) or zeros (0) – the sum of the column and row containing the COB will yield the length of each axis respectively. However, this method assumes that all other pixels not belonging to the RSO are zero (0).

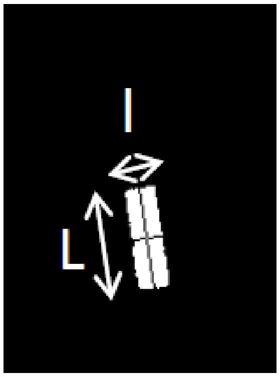


Figure 12: Major and Minor Axes of Blob

4.1.2.2.3. RSO Orientation Estimation

Given *a prioi* knowledge of the RSO, a numerical approximation for the projected area as a function of the ratio of the major-to-minor axis can be derived. For the case of a 3U Cubesat, **Error! Reference source not found.** illustrates the dataset from which the numerical approximations for the major and minor , given as Equation (16) and (17) respectively, are derived from. It is important to note that the area ratio is given with respect to the area of the smallest face of the RSO, A_0. For a 3U Cubesat, this corresponds to 100 cm^2.

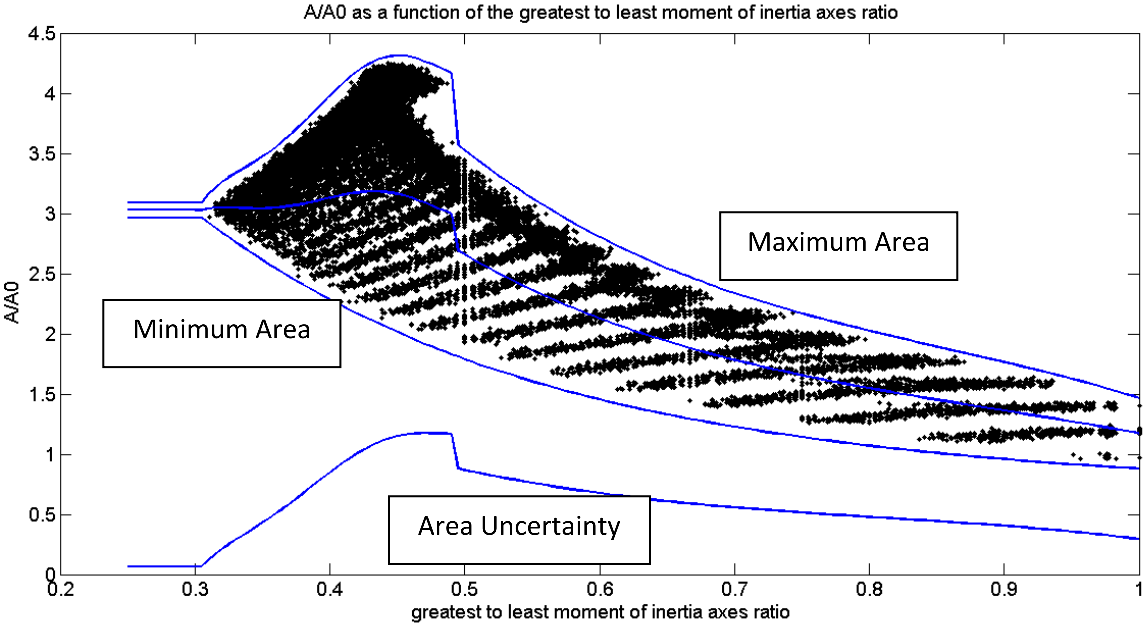


Figure 13. Projected Area as a Function of Axis Length Ratio

$$\frac{A}{A_0}(r)_{max} = \begin{cases} 3.091 & if\ 0.25 \le r < 0.305 \\ 71716r^5 - 144219r^4 - 114743r^3 - 45172r^2 + 8812.5r - 679.42 & if\ 0.305 \le r < 0.495 \\ 2.4853r^4 - 23.737r^3 + 50.09r^2 - 42.372r + 14.977 & if\ r \ge 0.495 \end{cases} \tag{16}$$

$$\frac{A}{A_0}(r)_{min} = \begin{cases} 2.963 & if\ 0.25 \le r < 0.305 \\ 5.51r^4 - 20.70r^3 + 30.73r^2 - 22.02r + 7.36 & if\ r \ge 0.305 \end{cases} \tag{17}$$

4.1.2.2.4. Range and Uncertainty Determination

Given the focal length of the lens, f, the area of each pixel, p^2, the number of pixels in the blob, *N*, and the numerical approximations for the projected, the average range, ρ, is given as

$$\rho = \frac{1}{2}\sqrt{\frac{A_0\left(\left(\frac{A}{A_0}\right)_{min} + \left(\frac{A}{A_0}\right)_{max}\right)}{Np^2}} \tag{18}$$

The uncertainty of the IPAs was shown to be primarily dependent upon the uncertainty in the area ratio function computed from the body axis ratio. The resulting estimated mean relative uncertainty is approximately 16% [17].

4.1.3. IPA Testing

4.1.3.1. Image Generation for Image Processing Algorithm Testing

Aiming towards the implementation of closed loop simulation of the AutoNav subsystem, a simulink-based image generator was required to mimic the input from the microbolometer into the IPAs. The image generator implementation was performed in two steps. The first one consisted of the implementation of a MATLAB code that can reliably generate a black and white image that accounts for location of the RSO within the FOV, inertial orientation of the RSO in space, and the distance between chief and deputy. The second part consisted of the actual integration of the image generator into Simulink so that the images being generated represent what the microbolometer would actually see in space, without consideration of Earth in the background or other possible satellites that the microbolometer might observe.

The MATLAB code that generates image matrices does this by representing the dimensions of the RSO and plotting them in a three-dimensional figure. Each of the individual faces is rotated with the Euler angles provided with a 3-2-1 rotation. The corresponding coordinates are then translated by a vectorial distance given as an input, which represents the location of the RSO within the FOV. To account for the range, all physical dimensions are multiplied by a scaling factor. The scaling factor is calculated as $\mathrm{s} = \frac{20}{\mathrm{d}_1}$, where $\mathrm{d}_1 = 2\rho \tan(\frac{\alpha}{2})$, corresponds to the total amount of units being plotted for the aspect ratio of the image to be precise, and $\alpha = 9.1^\circ$ is the angular aperture of the microbolometer's FOV. It is important to notice that the angle α corresponds to the aperture in the Y-axis in the ICF.

Next, the sides of the RSO are filled in. After rotating the view to an orientation that the imager will see, the aspect ratio is adjusted and the image is inversed (black to white and white to black) to create a representative image. It is possible to save a version of the image for reference, as well as to adding the Earth as a background by simple inclusion of one more input.

The second part of the open loop implementation of the IPA's is the integration into Simulink. The MATLAB code described above requires a call to the function coder.extrinsic() for Simulink to compile C-code related for plotting and some image processing functions. The various inputs to the Simulink image generator are: relative location of the RSO to Prox-1, orientation with respect to Prox-1, and the Euclidian range from Prox-1. The resulting Simulink block diagram is illustrated in Figure 14.

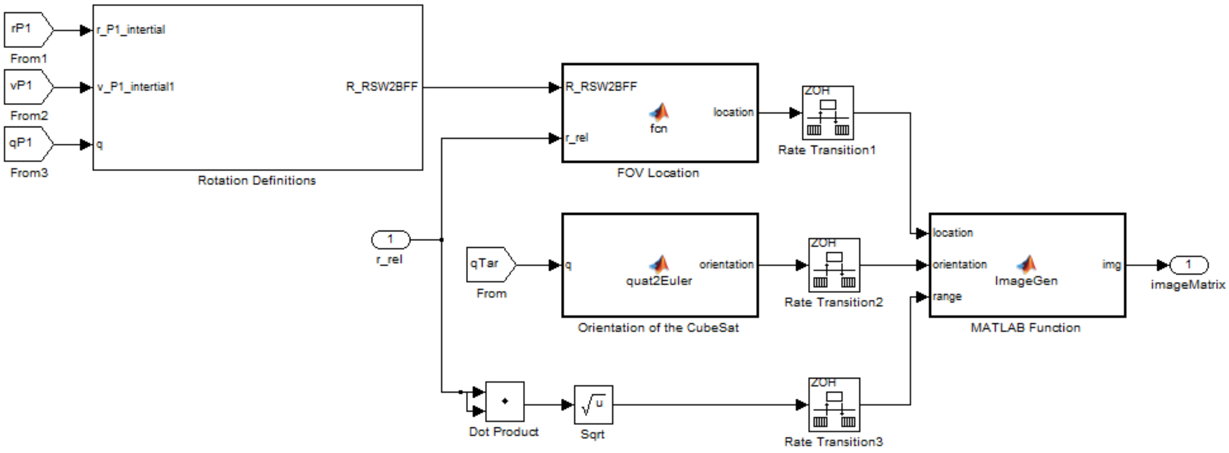


Figure 14. Simulink block diagram for image generator implementation.

4.1.3.2. Location Calculation

The location of the RSO within the FOV of the microbolometer is performed by expressing the relative position vector from Prox-1 to the RSO in the body fixed frame and projecting it upon the y-axis of the body fixed frame. Subtracting this projection from the relative position vector allows for the calculation of a vector projected upon the $\hat{X}_B - \hat{Z}_B$ plane.

4.1.3.3. Orientation Calculation

It is important to accurately describe the orientation of the RSO with respect to Prox-1 the uncertainty of the IPAs are dependent up on the area ratio calculation. The three Euler angles of interest are computed from the inertial quaternion of the RSO. The Euler angles used to perform the Eulerian 3-2-1 rotation are defined to be (in MATLAB notation)

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} atan2(\,2\cdot\ (q_0q_1 + q_2q_3)\,, 1 - 2\cdot(q_1^2 + q_2^2)\,) \\ \mathrm{asin}(\,2\cdot(q_0q_2 - q_3q_1)\,) \\ atan2(\,2\cdot\ (q_0q_3 + q_2q_1)\,, 1 - 2\cdot(q_3^2 + q_2^2)\,) \end{bmatrix} \tag{19}$$

The resulting rotation matrix to produce a 3-2-1 rotation is

$$R_{3-2-1} = R(\phi)R(\theta)R(\psi) \tag{20}$$

4.1.3.4. Covariance Matrix Determination

The covariance of the IPAs were analyzed and aided in determination of the measurement noise of the relative O/D filter. The inputs to the image generator were used as the truth data set. The covariance analysis contained a permutation of the three (3) Euler angles for the orientation of the RSO, where $\alpha_i \in \{0°, 45°, 90°\}$, as well as the location of the RSO in the FOV of the simulated imager. From this analysis, the standard deviation of the range error over all the orientations was obtained and is listed in Table 4 and illustrated graphically in Figure 15. Furthermore, Figure 16 illustrates the ratio of the error-to-range as a function of the range.

Table 4. Standard deviation of the error in range measurement as a function of rang

Range (m)	*1-σ Range Error (m)*
40	4.123
60	7.526
80	10.462
100	12.783
120	13.553
140	16.027

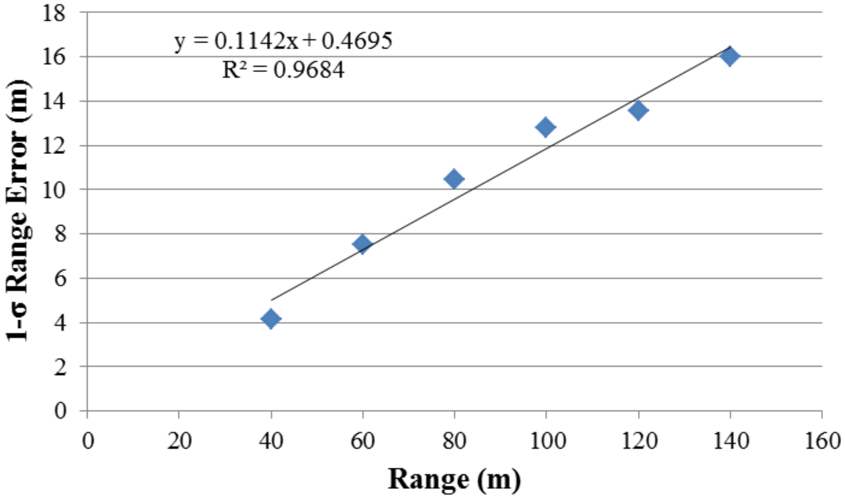


Figure 15. Standard deviation of the error in range measurement as a function of range.

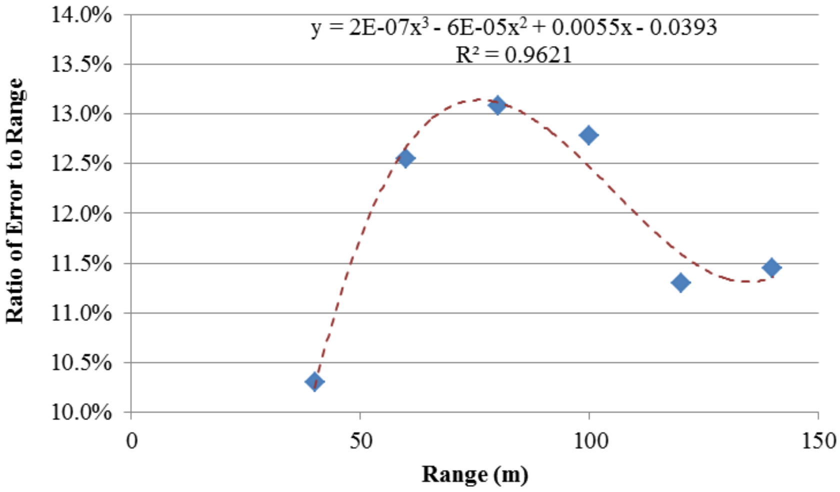


Figure 16. Ratio of $\frac{Range\ Error}{Range}$ *as a function of range.*

4.1.3.5. Image Processing Algorithm Boundary Analysis

In order to study the boundaries of the IPA's and their capability for detecting the RSO location under different orientations, three specific orientations were selected as illustrated by Figure 17.

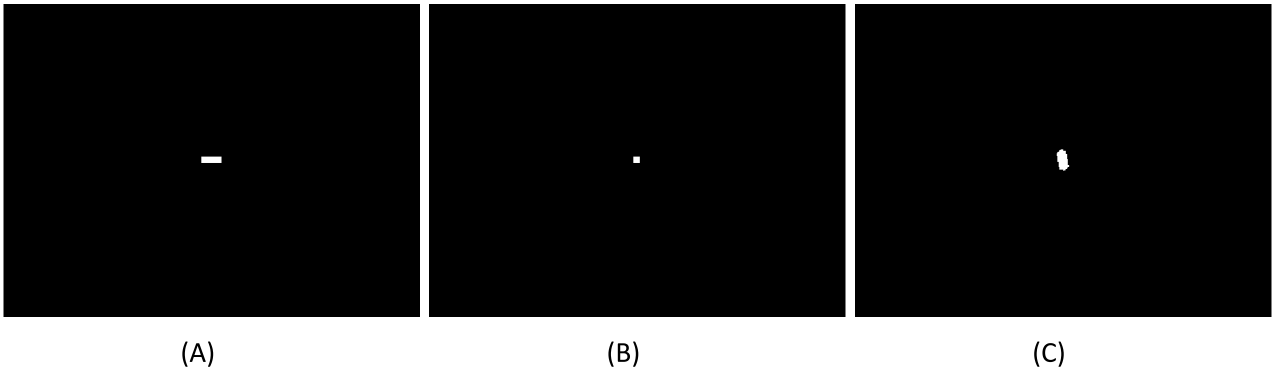


Figure 17. RSO orientations studied for image processing algorithm reliability. (A) Euler angles [ϕ,θ,ψ]=[0°,0°,0°]. (B) Euler angles [ϕ,θ,ψ]=[0°,0°,90°]. (C) Euler angles [ϕ,θ,ψ]=[45°,45°,45°].

These orientations expose the IPAs to both the minimum and maximum visible areas of the RSO. A simulation was created where every orientation was reproduced 24,400 different times for ranges between 30 and 150 m, in steps of 2 m, for 20 locations upon each of the axis. The intent was to cover the entire FOV of the simulated imager. From the output results, three metrics were used in determining whether a measurement was successful: (1) successfully found the the RSO (minimum number of pixels found), (2) correctly determined the aspect ratio to within a small error of the truth, and (3) the error in the range is less than the fitted error given as $0.1142x + 0.4695$.

Figure 18 illustrates the number of sample cases for which the IPAs failed to provide an accurate estimate of the location of the RSO as a function of true range in meters. As expected, the orientation $\{0°, 0°, 0°\}$ provided the highest reliability with 85% success. This is the case for which the IPA's is most finely tuned, since the CubeSat is represented by its exact aspect ratio. On the other hand, the cases for $\{45°, 45°, 45°\}$ and $\{0°, 0°, 90°\}$ provided a low success rate. The success rate for these was 32.75% and 25.26% respectively. The failure distribution $\{45°, 45°, 45°\}$ test case is a bimodal distribution a minima occurring between the ranges of 100m and 110m.

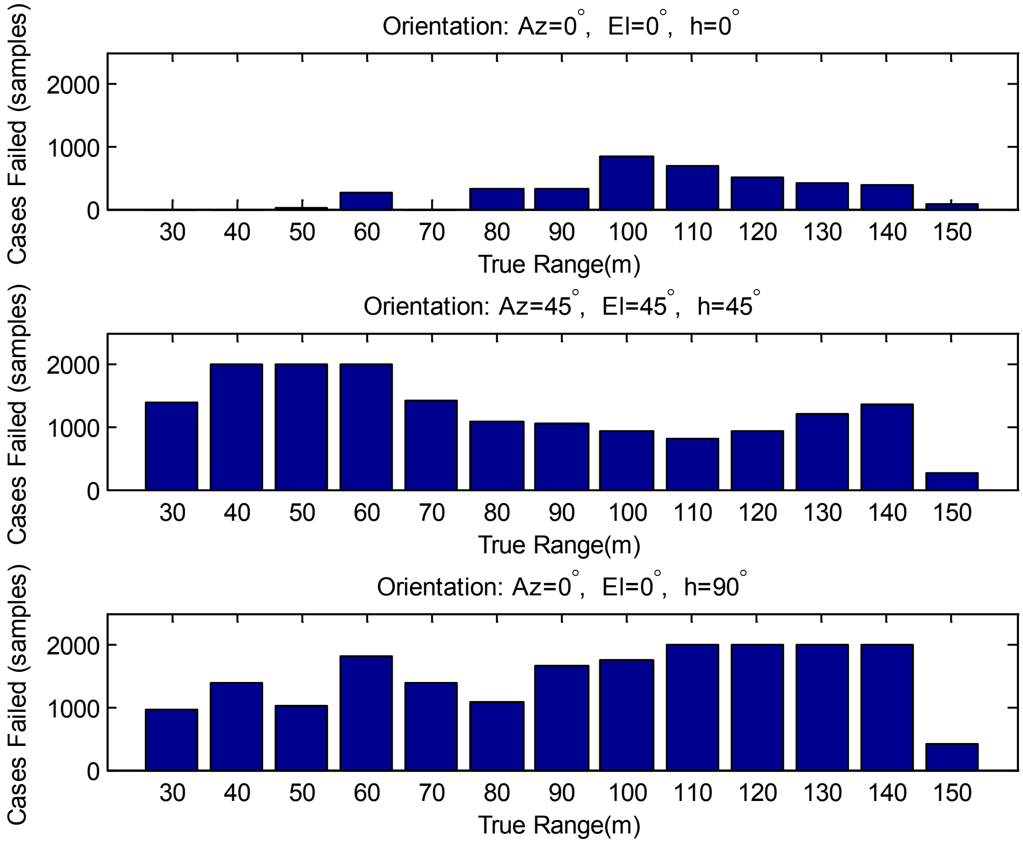


Figure 18.Number of cases for which the IPA's failed to approximate range to CubeSat as a function of range, for different orientations.

4.2. *Artificial Potential Functions*

4.2.1. Overview

Compared to other guidance control algorithms that utilize a Linear Quadratic Regulator (LQR) controller, for example, artificial potential functions (APFs) have a much lower computation cost as they can consist of only arithmetic operators – i.e. addition, subtraction, multiplication, division. Additionally, the underlying formulation of the APFs includes the use of attractors and repellers such that a global minima is created. A basic overview of attractors and repellers are discussed for informational purposes.

4.2.2. Attractors and Repellers

Let $p \epsilon \mathbb{R}$ be a fixed point such that

$$f(p) = p \tag{21}$$

If $|f'(p)| < 1$, then p is classified as an attractor or an attractive fixed point. If $f'(p) = 0, p$ is said to be super-attractive or super-stable. Furthermore, for points that are sufficiently close to an attractive fixed point converge geometrically [19]. In other words, let p be a super-stable fixed point. Then, $f(p + \delta)$, where $\delta < \varepsilon$ as illustrated by Figure 19, can be shown to converge back to p since $|f'(p + \delta)| < 1$.

Conversely, if $|f'(p)| > 1$, then p is classified as a repeller or a repelling fixed point. Like an attractive fixed point, points sufficiently close can be shown to diverge from the point p.

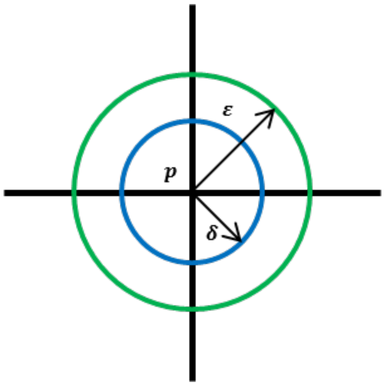


***Figure 19. The attractive basin around the fixed point* p.**

Expanding upon this idea, the attractive potential function must have either an attractive, or super-attractive fixed point at the goal. By choosing the APFs such that they are valid Lyanpunov candidate functions (LCF) will ensure that the goal is a fixed point and at the global minimum.

4.2.3. APF Guidance Formulation

4.2.3.1. Overview

In order to use the APFs as a feasible guidance formulation, it must exhibit a global minimum at the goal (attractor) and regions of high potential around obstacles and/or keep out zones (repellers). The total potential function is defined as the superposition of the sum of the attractors and the sum of the repellers,

$$\underline{\boldsymbol{\phi}} = \underline{\boldsymbol{\phi}}_A + \underline{\boldsymbol{\phi}}_R \tag{22}$$

To assure a solution is achieved, the APFs are treated as a Lyapunov candidate function (LCF) [20]. In order for a candidate function to valid, it must meet the following criteria:

$$\begin{array}{rlll} 1. & \underline{\boldsymbol{\phi}}(\boldsymbol{r}, t) > 0 & \forall & \boldsymbol{r} \neq 0 \\ & \underline{\boldsymbol{\phi}}(\boldsymbol{r}, t) = 0 & for & \boldsymbol{r} = 0 \\ 2. & \dot{\underline{\boldsymbol{\phi}}}(\boldsymbol{r}, \dot{\boldsymbol{r}}, t) < 0 & \forall & \boldsymbol{r}, \dot{\boldsymbol{r}} \rightarrow 0 \end{array} \tag{23}$$

4.2.3.2. Attractive Potential

The attractive potential, $\underline{\boldsymbol{\phi}}_A$is defined to be

$$\underline{\boldsymbol{\phi}}_A = \frac{\kappa_A}{2}(\boldsymbol{r} - \boldsymbol{r}^*)^{\boldsymbol{T}} \underline{\boldsymbol{Q}}_{\boldsymbol{A}} (\boldsymbol{r} - \boldsymbol{r}^*) \tag{24}$$

where $(\boldsymbol{r} - \boldsymbol{r}^*)$ is the relative position of the chaser spacecraft to the goal in the RSW frame, or desired position, $\kappa_A \in \mathbb{R}^1$ is a small positive gain, and $\underline{\boldsymbol{Q}}_{\boldsymbol{A}} \in \mathbb{R}^{3x3}$ is a positive definite shaping matrix for the attractive potential. Figure 20 illustrates effects of the shaping matrix $\underline{\boldsymbol{Q}_A}$ and gain κ_a on the attractive potential. For this illustration, Figure 20(a) illustrates the baseline APF with the shaping matrix set to the Identity matrix. Significant geometric changes occur when the radial component of the shaping matrix is altered. As illustrated by Figure 20(c), when the radial component of the shaping matrix is an order of magnitude larger than the in-track component, the attractive potential field becomes more bowl-shaped. Changing the in-track component of the shaping matrix is equivalent to scaling the gain by the same amount while holding the shaping matrix equal to the identity matrix, $\underline{\boldsymbol{Q}_A} = \boldsymbol{I}$. Cross-track changes affect neither the geometry nor the scaling of the attractive potential field.

4.2.3.3. Repulsive Potential

The repulsive potential, $\underline{\boldsymbol{\phi}}_R$, is defined to be

$$\underline{\boldsymbol{\phi}}_R = \frac{\kappa_R}{2}\frac{(\boldsymbol{r}-\boldsymbol{r}^*)^T\underline{\boldsymbol{Q}_A}(\boldsymbol{r}-\boldsymbol{r}^*)}{(\boldsymbol{r}-\boldsymbol{r}^\Delta)^T\underline{\boldsymbol{P}_A}(\boldsymbol{r}-\boldsymbol{r}^\Delta)-1} \tag{25}$$

where $(\boldsymbol{r}-\boldsymbol{r}^\Delta)$ is the relative position of the chaser spacecraft to the RSO, $\kappa_R \epsilon \mathbb{R}^1$ is a small positive gain, and $\underline{\boldsymbol{P}_A} \epsilon \mathbb{R}^{3x3}$ is a positive definite shaping matrix for the repulsive potential. This shaping can be used to virtually encapsulate the RSO in a 'Keep Out Region'. The 'Keep Out Region' is enforced mathematical a sudden increase in the repulsive potential as the chaser spacecraft encroaches upon the virtual ellipsoidal projection that surrounds the RSO.

4.2.3.4. Control Law Formulation

In order to ensure $\underline{\boldsymbol{\phi}}$ is a valid LCF, $\dot{\underline{\boldsymbol{\phi}}}$ must be negative definite. Taking the time derivative of the total potential yields

$$\dot{\underline{\boldsymbol{\phi}}} = \dot{\underline{\boldsymbol{\phi}}}_A + \dot{\underline{\boldsymbol{\phi}}}_R = \nabla\underline{\boldsymbol{\phi}} \tag{26}$$

Examining the attractive potential first,

$$\frac{\partial \underline{\boldsymbol{\phi}}_A}{\partial \boldsymbol{r}} = \frac{\partial}{\partial \boldsymbol{r}}\left(\frac{\kappa_A}{2}\left[\boldsymbol{r}^T\underline{\boldsymbol{Q}_A}\boldsymbol{r} - \boldsymbol{r}^T\,\underline{\boldsymbol{Q}_A}\boldsymbol{r}^* - \boldsymbol{r}^{*T}\underline{\boldsymbol{Q}_A}\boldsymbol{r} + \boldsymbol{r}^{*T}\underline{\boldsymbol{Q}_A}\boldsymbol{r}^*\right]\right) \tag{27}$$

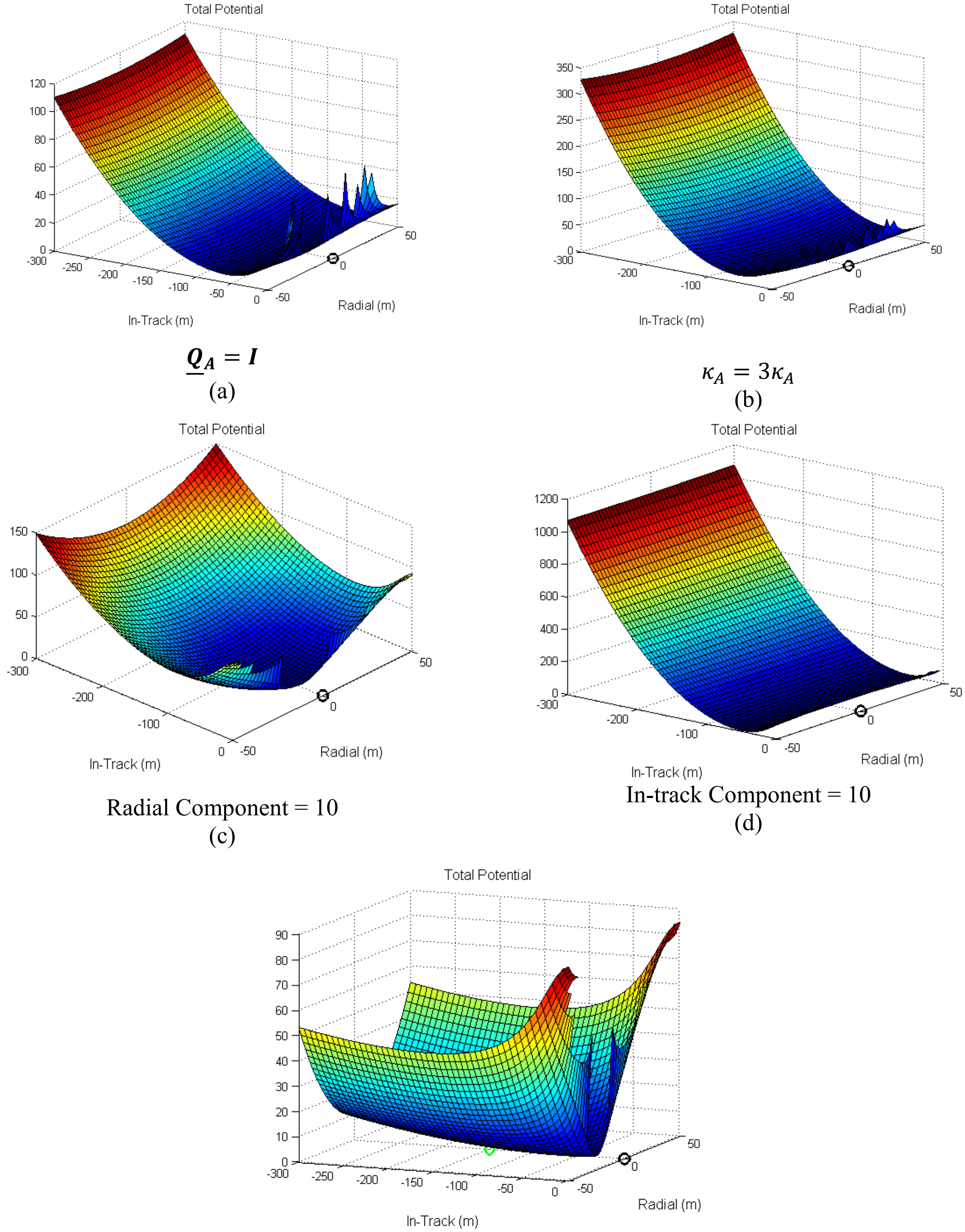
Taking the partial derivative and collecting like terms,

$$\frac{\partial \underline{\boldsymbol{\phi}}_A}{\partial \boldsymbol{r}} = \frac{\kappa_A}{2}\left[\boldsymbol{r}^T\underline{\boldsymbol{Q}_A} + \underline{\boldsymbol{Q}_A}\boldsymbol{r} - \underline{\boldsymbol{Q}_A}\boldsymbol{r}^* - \boldsymbol{r}^{*T}\underline{\boldsymbol{Q}_A}\right] \tag{28}$$

Since $\underline{\boldsymbol{Q}_A} = \underline{\boldsymbol{Q}_A^T}$, it follows that $\boldsymbol{r}^T\underline{\boldsymbol{Q}_A} = \underline{\boldsymbol{Q}_A}\boldsymbol{r}$. Therefore, Equation (28) becomes

$$\nabla\underline{\boldsymbol{\phi}}_A = \kappa_A\underline{\boldsymbol{Q}_A}\boldsymbol{r}_{CT} \tag{29}$$

where $\boldsymbol{r}_{CT} = (\boldsymbol{r}-\boldsymbol{r}^*)$ is the relative position of the chaser spacecraft to the desired (or targeted) position.



$\underline{\boldsymbol{Q}}_A = \boldsymbol{I}$

(a)

$\kappa_A = 3\kappa_A$

(b)

Radial Component = 10

(c)

In-track Component = 10

(d)

Radial Component = 10

In-track Component = 0.1

(d)

Figure 20. Effects of the shaping matrix $\underline{Q}_A$ on the attractive potential

Next, following a similar procedure, the gradient of the repulsive potential is

$$\nabla\underline{\boldsymbol{\phi}}_R = \kappa_R \frac{(\boldsymbol{r}_{CD}^T \underline{\boldsymbol{P}}_{\boldsymbol{A}} \boldsymbol{r}_{CD} - 1)\underline{\boldsymbol{Q}}_{\boldsymbol{A}} - \boldsymbol{r}_{CT}^T \underline{\boldsymbol{Q}}_{\boldsymbol{A}} \boldsymbol{r}_{CT}(\underline{\boldsymbol{P}}_{\boldsymbol{A}} \boldsymbol{r}_{CD})}{(\boldsymbol{r}_{CD}^T \underline{\boldsymbol{P}}_{\boldsymbol{A}} \boldsymbol{r}_{CD} - 1)^2} \tag{30}$$

where $\boldsymbol{r}_{CD} = (\boldsymbol{r} - \boldsymbol{r}^{\Delta})$ is the relative position of the chaser spacecraft to the RSO (or deputy spacecraft).

Therefore, the gradient of the total potential becomes

$$\nabla\underline{\boldsymbol{\phi}} = \kappa_A \underline{\boldsymbol{Q}}_{\boldsymbol{A}} \boldsymbol{r}_{CT} + \kappa_R \frac{(\boldsymbol{r}_{CD}^T \underline{\boldsymbol{P}}_{\boldsymbol{A}} \boldsymbol{r}_{CD} - 1)\underline{\boldsymbol{Q}}_{\boldsymbol{A}} - \boldsymbol{r}_{CT}^T \underline{\boldsymbol{Q}}_{\boldsymbol{A}} \boldsymbol{r}_{CT}(\underline{\boldsymbol{P}}_{\boldsymbol{A}} \boldsymbol{r}_{CD})}{(\boldsymbol{r}_{CD}^T \underline{\boldsymbol{P}}_{\boldsymbol{A}} \boldsymbol{r}_{CD} - 1)^2} \tag{31}$$

In order to meet the negative definite requirement, the negative gradient of the total potential is taken

$$-\nabla\underline{\boldsymbol{\phi}} = \boldsymbol{V}_D = -\kappa_A \underline{\boldsymbol{Q}}_{\boldsymbol{A}} \boldsymbol{r}_{CT} - \kappa_R \frac{(\boldsymbol{r}_{CD}^T \underline{\boldsymbol{P}}_{\boldsymbol{A}} \boldsymbol{r}_{CD} - 1)\underline{\boldsymbol{Q}}_{\boldsymbol{A}} - \boldsymbol{r}_{CT}^T \underline{\boldsymbol{Q}}_{\boldsymbol{A}} \boldsymbol{r}_{CT}(\underline{\boldsymbol{P}}_{\boldsymbol{A}} \boldsymbol{r}_{CD})}{(\boldsymbol{r}_{CD}^T \underline{\boldsymbol{P}}_{\boldsymbol{A}} \boldsymbol{r}_{CD} - 1)^2} \tag{32}$$

where $\boldsymbol{V}_D \in \mathbb{R}^3$ is the desired velocity at any point along the total potential field in the RSW frame [20, 21].

The control algorithm used to implement the APF guidance formulation drives the error velocity to zero (0), as illustrated in Figure 21. For a rest-to-rest maneuver, the vector $\boldsymbol{V}_{target}$ is zero (0) – assuming the desired position is not moving – and the error velocity becomes the velocity of the chaser in the RSW frame.

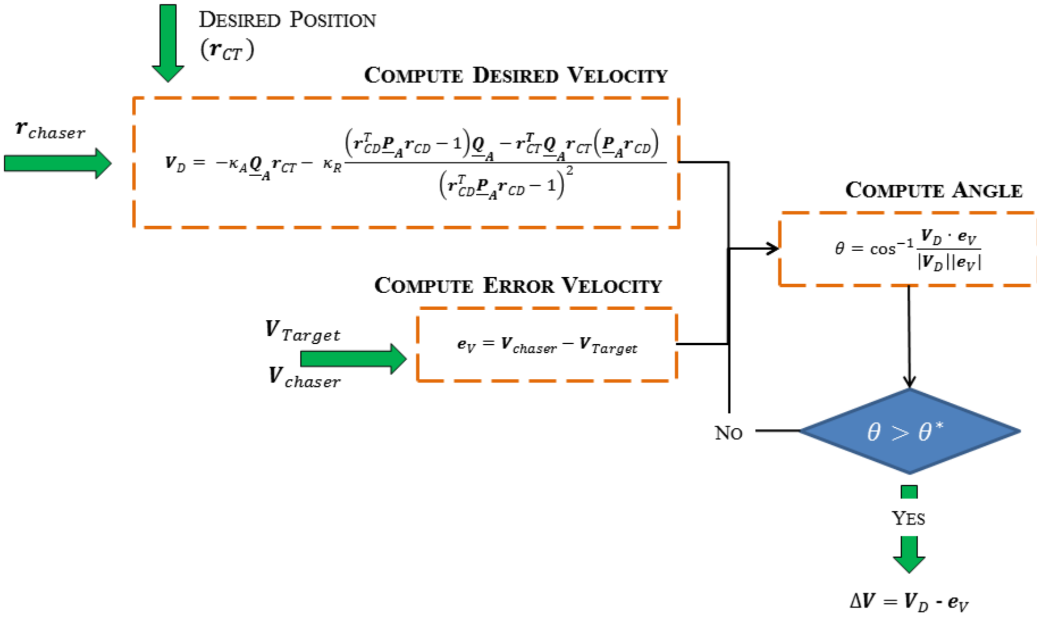

Desired Position
$(\boldsymbol{r}_{CT})$
Compute Desired Velocity
$\boldsymbol{r}_{chaser}$
$\boldsymbol{V}_D = -\kappa_A \underline{\boldsymbol{Q}}_A \boldsymbol{r}_{CT} - \kappa_R \frac{(\boldsymbol{r}_{CD}^T \underline{\boldsymbol{P}}_A \boldsymbol{r}_{CD} - 1)\underline{\boldsymbol{Q}}_A - \boldsymbol{r}_{CT}^T \underline{\boldsymbol{Q}}_A \boldsymbol{r}_{CT}(\underline{\boldsymbol{P}}_A \boldsymbol{r}_{CD})}{(\boldsymbol{r}_{CD}^T \underline{\boldsymbol{P}}_A \boldsymbol{r}_{CD} - 1)^2}$
Compute Angle
$\theta = \cos^{-1}\frac{\boldsymbol{V}_D \cdot \boldsymbol{e}_V}{|\boldsymbol{V}_D||\boldsymbol{e}_V|}$
Compute Error Velocity
$\boldsymbol{V}_{Target}$
$\boldsymbol{V}_{chaser}$
$\boldsymbol{e}_V = \boldsymbol{V}_{chaser} - \boldsymbol{V}_{Target}$
No
$\theta > \theta^*$
Yes
$\Delta\boldsymbol{V} = \boldsymbol{V}_D$ - $\boldsymbol{e}_V$

Figure 21. APF Guidance controller

4.2.4. APF Guidance Strategy and Preliminary Results

4.2.4.1. Rest-to-Rest Maneuver

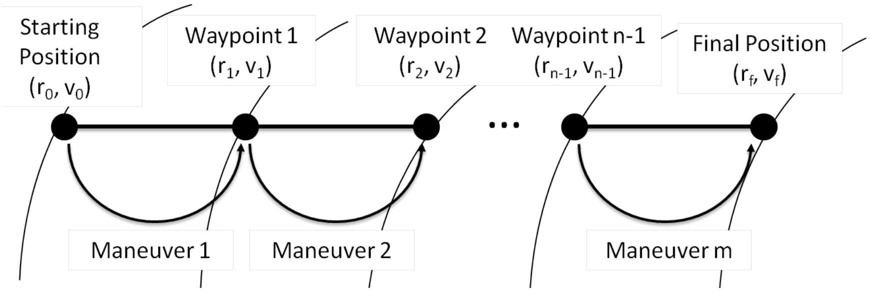


Figure 22: Waypoint-based maneuver diagram.

During formation flying, Prox-1 will utilize a waypoint-based maneuver strategy to gradually approach the RSO, as illustrated in Figure 22. The intent of this strategy is to avoid an possible collision with the RSO. The APF implementation will require an updated relative orbital determination (O/D) solution throughout the approach. While, given a waypoint-based maneuver strategy is not necessary, it mitigates risk in two (2) distinct ways: (1) reduces the time of flight (TOF) of any leg of the approach/retreat; and (2) decreases the chances of overshooting the desired goal position and colliding with the RSO.

Figure 23 illustrates the results of a rest-to-rest maneuver from a 120m trailing orbit to 50m (stable) trailing orbit with no intermediate waypoints. Performing the maneuver consumes approximately 6.6g of fuel – just under 2% of the total on-board propellant capacity.

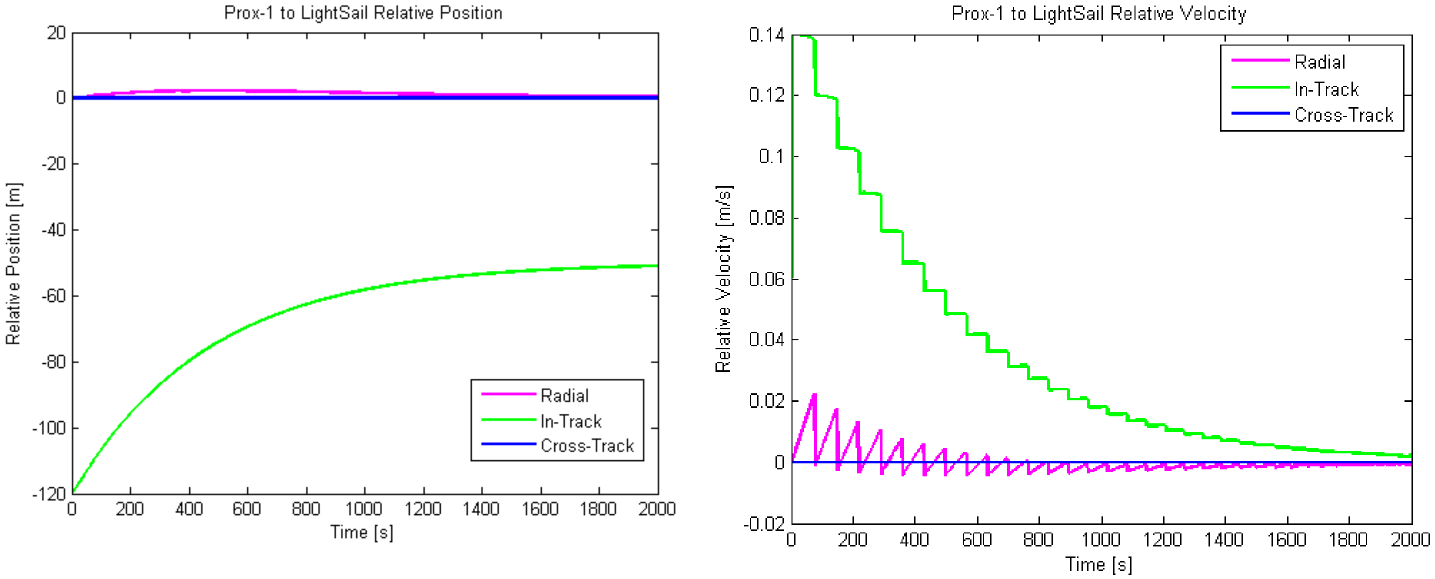


Figure 23. Results of a rest-to-rest maneuver a 120m to 50m trailing orbit with no intermediate waypoints.

4.2.4.2. Natural Motion Circumnavigation (NMC)

The relative orbital dynamics, as demonstrated by the CW equations illustrate in Figure 24, indicate that, given a correct initial state, a natural motion relative orbit can be achieved around a target. In the ideal scenario a NMC can be entered with a single burn should the initial relative position of the chaser lie on the desired orbit. This NMC could then be carried out indefinitely, sans external disturbances such as solar ration pressure (SRP), drag, et cetera, without any thruster control, only requiring slew maneuvers to track the target spacecraft. In this case the required ΔV for NMC insertion would be minimal, as listed in Table 5.

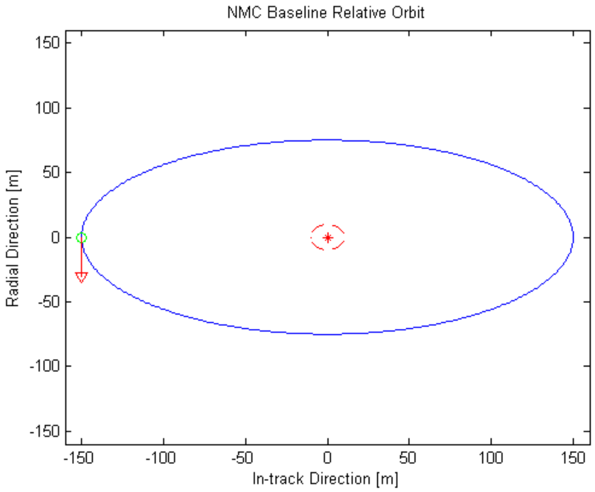


Figure 24. Natrual Motion Circumnavigation (NMC) about the RSO.

Table 5. NMC NominalΔV requirements

Semi-Major Axis (m)	*ΔV (m/s)*
100	0.0553
236	0.0692
150	0.0830
175	0.0986
200	0.1107

An equal in magnitude, opposite in direction burn would be required when the chaser returns to its original position exactly one orbital period after insertion to return to the original stationary trailing orbit.

However, should an incorrect thrust maneuver be performed or natural orbital perturbations cause a significant change in spacecraft relative velocity, the chaser will drift out of its desired NMC. Error analysis shows that an incorrect applied velocity vector of 0.01 m/s has the potential to cause the Prox-1 spacecraft to enter the keep out region surrounding the RSO within one orbit if left uncontrolled. It is therefore necessary to develop a guidance strategy to inject Prox-1 into a correct NMC and should an incorrect injection or unexpected change in velocity occur, re-insert Prox-1 into a stable relative orbit.

The guidance strategy will be based upon a dynamic application of the rest-to-rest APF control law. During NMC, the guidance algorithms will track two error terms, each of which are capable of triggering a burn should they exceed their allowable error tolerances. First, the guidance algorithm tracks the current relative position of Prox-1 versus the current relative position vector as determined by the propagation of the original desired NMC. If this relative error becomes too large, a maneuver flag will become tripped within the guidance system initiating the corrective guidance algorithm.

The guidance algorithm will also track the relative velocity of Prox-1 versus the desired velocity as determined by the propagated reference NMC. Should the relative velocity error become large enough where the guidance system deems an undesired state will result with time, a maneuver flag is also tripped.

When it is determined that a corrective maneuver is required, a two-fold approach will be used to correct the orbit. First, APF's will be utilized to drive the chaser towards a reference NMC, and second, the CW equations will be used to correct the relative velocity and re-insert Prox-1 along the reference NMC.

Since the APF control law is derived using a chaser satellite that is driving towards a specified target, a specific position on the NMC must be chosen that is ahead of the Prox-1 spacecraft along some reference NMC. This dynamically updated desired will be used to drive Prox-1 towards it should the guidance algorithm deem it the error is too large and drive Prox-1 back to the reference NMC. After the APFs drive Prox-1 into a position on the reference NMC, the CW equations are used to calculate the necessary velocity to maintain circumnavigation and the necessary thrust applied. Once Prox-1 is determined to be within the acceptable error parameters, no thrust maneuvers occur.

A baseline test case was performed whose results are shown in Figure 25. The test case was initialized with an incorrect initial position and velocity and was able to successfully demonstrate the validity of the APF NMC guidance. The ΔV requirements of this formulation, in a worst case scenario have also shown to be acceptable with a use of approximately 1.1 m/s per NMC. Although in the early stages of development, this guidance strategy has shown great promise and further refinement should yield significantly better results and control authority.

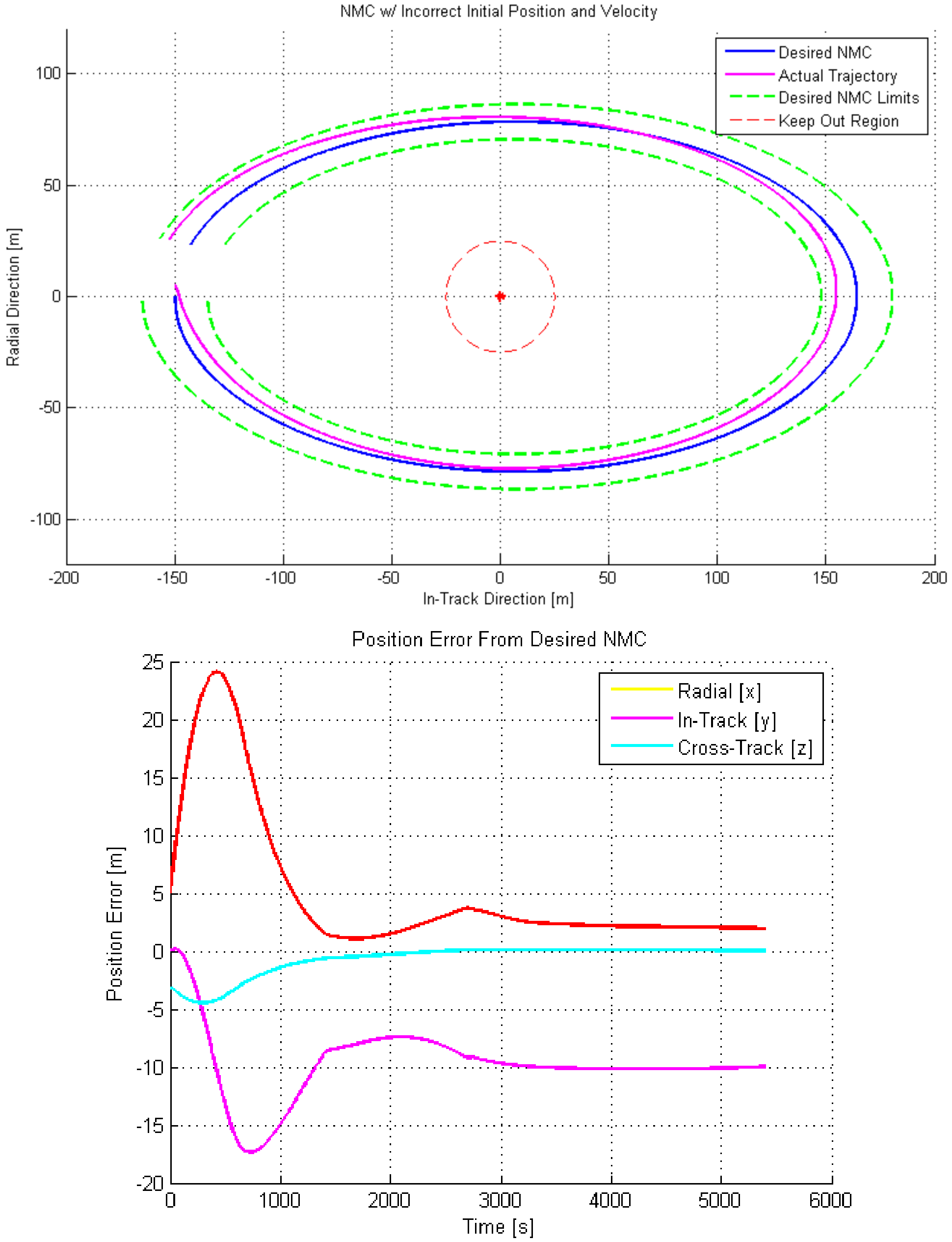


Figure 25. Results of baseline algorithm testing.

5. Navigation Formulations

5.1. *Overview*

The 'Navigation' portion of the GN&C refers to the determination of the vehicle's state, classically including position, velocity, and attitude. Classical methods of determining position and velocity include integrating the equations of motion based on sensor inputs from an Inertial Measurement Unit (IMU). More modern methods include the use of various measurement methods, such as GPS, magnetometers, rate gyros, accelerometers, et cetera, and blend them together via sensor fusion. This method typically involves the use a filter – typically a variant of the Kalman Filter (such as the Kalman Filter, Extended Kalman Filter, or the Unscented Kalman Filter) – to produce an optimal state estimate.

5.2. *Filter Derivation*

5.2.1. Overview

The Kalman Filter (KF) is an optimal sequential state estimator which is derived from the models of stochastic dynamical systems in addition to stochastic models of sensor measurements. The use of the measurement model is an integral part of the filtering process as it relates measurements from several different sensors to the state variables in the filter – thus allowing a spectrum of various measurement methods to be used. To account for nonlinearities in the both the dynamics and measurement models, an Extended Kalman Filter (EKF) can be utilized. The EKF differs in several ways from the KF, but the basic algorithm between the two versions is similar:

1) State Prediction
2) Compute Prediction Error
3) Compute Optimal Gain
4) Update State Estimate
5) Compute Error for Updated State Estimate

The next two (2) sections will first derive the Kalman filter and then introduce modifications of the Kalman filter to account for the nonlinearities of a dynamics and measurement model.

5.2.2. Kalman Filter Derivation

5.2.2.1. State Prediction

Let $x \epsilon \mathbb{R}^n$ be a Guass-Markov process such that an initial condition, x_0, is known and

$$x_k = \underline{\boldsymbol{\phi}}_{k-1} x_{k-1} + w_{k-1} \tag{33}$$

where $\underline{\boldsymbol{\phi}} \epsilon \mathbb{R}^{n \, x \, n}$ is the state transition matrix (STM) and $w_{k-1} \epsilon \mathbb{R}^n$ is an independent and identically distributed (iid) random variable, called the process noise. The process noise accounts for unmodeled disturbances acting on the system and is assumed to be distributed normally with zero (0) mean and strength $\underline{\boldsymbol{Q}}$. Assuming the estimate $\hat{x}_{k-1}$ exists for time *k-1*, then the prediction of the state at time *k* is

$$\hat{x}_{k(-)} = \underline{\boldsymbol{\phi}}_{k-1}\hat{x}_{k-1} \tag{34}$$

5.2.2.2. Prediction Error Covariance

The error covariance matrix $\underline{\boldsymbol{P}}_{k(-)}$ is given as

$$\underline{\boldsymbol{P}}_{k(-)} = E\left\{\left(x_k - \hat{x}_{k(-)}\right)^T\left(x_k - \hat{x}_{k(-)}\right)\right\} \tag{35}$$

Defining

$$\begin{aligned} e_{k(-)} &\equiv \hat{x}_{k(-)} - x_k \\ &= \underline{\boldsymbol{\phi}}_{k-1}\hat{x}_{k-1} - x_k \\ &= \underline{\boldsymbol{\phi}}_{k-1}\hat{x}_{k-1} - \underline{\boldsymbol{\phi}}_{k-1}x_{k-1} - w_{k-1} \\ &= \underline{\boldsymbol{\phi}}_{k-1}(\hat{x}_{k-1} - x_{k-1}) - w_{k-1} \\ &= \underline{\boldsymbol{\phi}}_{k-1}e_{k-1} - w_{k-1} \end{aligned} \tag{36}$$

Therefore,

$$E\left\{e_{k(-)}e_{k(-)}^T\right\} = \underline{\boldsymbol{\phi}}_{k-1}E\{e_{k-1}e_{k-1}^T\}\underline{\boldsymbol{\phi}}_{k-1}^T + \underline{\boldsymbol{Q}}_{k-1} \tag{37}$$

where $\underline{\boldsymbol{Q}}_{k-1} = E\{w_{k-1}w_{k-1}^T\} \epsilon\, \mathbb{R}^{n\,\mathrm{x}\,n}$ be a positive semi-definite symmetric matrix and the prediction covariance is

$$\underline{\boldsymbol{P}}_{k(-)} = \underline{\boldsymbol{\phi}}_{k-1}\underline{\boldsymbol{P}}_{k-1}\underline{\boldsymbol{\phi}}_{k-1}^T + \underline{\boldsymbol{Q}}_{k-1} \tag{38}$$

5.2.2.3. Kalman Gain, State Estimate Update

Consider the following cost function at step *k*,

$$J\left(x_k, \hat{x}_{k(-)}\right) = \frac{1}{2}\left(\boldsymbol{z}_k - \underline{\boldsymbol{H}}_k x_k\right)^T \underline{\boldsymbol{R}}_k^{-1}\left(\boldsymbol{z}_k - \underline{\boldsymbol{H}}_k x_k\right) + \frac{1}{2}\left(x_k - \hat{x}_{k(-)}\right)^T \underline{\boldsymbol{P}}_{k(-)}^{-1}\left(x_k - \hat{x}_{k(-)}\right) \tag{39}$$

where $\boldsymbol{z}_k \,\epsilon\, \mathbb{R}^m$ is the measurement vector, $\underline{\boldsymbol{H}}_k \,\epsilon\, \mathbb{R}^{m\,\mathrm{x}\,n}$ is the linear measurement model, and $\underline{\boldsymbol{R}}_k = E\{\eta_k\eta_k^T\} \,\epsilon\, \mathbb{R}^{m\,\mathrm{x}\,m}$ is the measurement noise. It can be shown that the updated state estimate, $\hat{x}_k$, is

$$\hat{x}_k = \hat{x}_{k(-)} + \underline{\boldsymbol{K}}_k(\boldsymbol{z}_k - \underline{\boldsymbol{H}}_k\hat{x}_{k(-)}) \tag{40}$$

where $\underline{\boldsymbol{K}}_k \,\epsilon\, \mathbb{R}^{n\,\mathrm{x}\,m}$ is the Kalman Gain and is given as

$$\underline{\boldsymbol{K}}_k = \underline{\boldsymbol{P}}_{k(-)}\underline{\boldsymbol{H}}_k^T\left(\underline{\boldsymbol{H}}_k\underline{\boldsymbol{P}}_{k(-)}\underline{\boldsymbol{H}}_k^T - \underline{\boldsymbol{R}}_k\right)^{-1} \tag{41}$$

minimizes the cost function in Equation (39) - but is too tedious to do here as it is crashing Word.

5.2.2.4. State Prediction Error Update

Beginning with the State Update Equation, Equation (40), defining $e_k = \hat{x}_k - x_k$,

$$e_k = \hat{x}_{k(-)} - x_k + \underline{\boldsymbol{K}}_k\left(\boldsymbol{z}_k - \underline{\boldsymbol{H}}_k x_k - \underline{\boldsymbol{H}}_k\left(\hat{x}_{k(-)} - x_k\right)\right) \tag{42}$$

Since the measurement equation is given as $z_k = \underline{\boldsymbol{H}}_k x_k + \eta_k$

$$\begin{aligned} e_k &= e_{k(-)} + \underline{\boldsymbol{K}}_k\left(\eta_k - \underline{\boldsymbol{H}}_k e_{k(-)}\right) \\ &= \left(\boldsymbol{I} - \underline{\boldsymbol{K}}_k\underline{\boldsymbol{H}}_k\right)e_{k(-)} + \underline{\boldsymbol{K}}_k\eta_k \end{aligned} \tag{43}$$

Then,

$$E\{e_k e_k^T\} = \left(\boldsymbol{I} - \underline{\boldsymbol{K}}_k\underline{\boldsymbol{H}}_k\right) E\{e_{k(-)}e_{k(-)}^T\}\left(\boldsymbol{I} - \underline{\boldsymbol{K}}_k\underline{\boldsymbol{H}}_k\right)^T + \underline{\boldsymbol{K}}_k\underline{\boldsymbol{R}}_k\underline{\boldsymbol{K}}_k^T \tag{44}$$

and

$$\underline{\boldsymbol{P}}_k = \left(\boldsymbol{I} - \underline{\boldsymbol{K}}_k\underline{\boldsymbol{H}}_k\right) \underline{\boldsymbol{P}}_{k(-)}\left(\boldsymbol{I} - \underline{\boldsymbol{K}}_k\underline{\boldsymbol{H}}_k\right)^T + \underline{\boldsymbol{K}}_k\underline{\boldsymbol{R}}_k\underline{\boldsymbol{K}}_k^T \tag{45}$$

It is important to note that this form of the error covariance matrix is called the *Joseph Form*. Compared to other equivalent forms of the error covariance matrix, the Joseph Form provides improved numerical stability and precision and guarantees the matrix is symmetric and positive definite.

5.2.3. Extended Kalman Filter Extension

5.2.3.1. State Prediction

For a Kalman Filter, the (linearly) predicted state is simply the previous state transformed to the new state via a state transition matrix. For the non-linear case, the state transition matrix can simply be replaced by the non-linear state equation with no process noise. The resulting state prediction is

$$\hat{x}_{k_-} = f(x_{k-1}) \tag{46}$$

5.2.3.2. Prediction Error Covariance

In order to capture the non-linear dynamics, $f(x_{k-1}, \ldots)$ can be expanding in a Taylor series about the point $\hat{x}_{k-1}$ yielding

$$f(x_{k-1}, \ldots) = f(x_{k-1}) + \underline{\boldsymbol{J}}_x \big|_{\hat{x}_{k-1}}(x_{k-1} - \hat{x}_{k-1}) \tag{47}$$

where $J_{\hat{x}_{k-1}} \epsilon \, \mathbb{R}^{n \, x \, n}$ is the Jacobian of $f(\cdot)$ evaluated at the point $\hat{x}_{k-1}$. Additionally, the error covariance matrix $P_{k\_}$ can be defined in an analogous way to the KF

$$P_{k\_} = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \tag{48}$$

The error, defined as,

$$e_{k\_} \equiv \hat{x}_{k\_} - x_k \ = f(\hat{x}_{k-1}) + \omega_{k-1} - f(x_{k-1}) \tag{49}$$

becomes

$$e_{k\_} = \ f(\hat{x}_{k-1} - x_{k-1} \,) + \omega_{k-1} \tag{50}$$

which is approximately

$$e_{k\_} \approx J_{\hat{x}_{k-1}} e_{k-1} + \ \omega_{k-1} \tag{51}$$

The error covariance $P_{k\_}$ becomes

$$E\left[e_{k\_} \; e_{k\_}^T\right] = \underline{\boldsymbol{J}}_x \,|_{\,\hat{x}_{k-1}} E[e_{k-1} e_{k-1}^T] \, \underline{\boldsymbol{J}}_x^T \,|_{\,\hat{x}_{k-1}} \ + E[\omega_{k-1} \; \omega_{k-1}^T] \tag{52}$$

It follows that,

$$P_{k\_} = \underline{\boldsymbol{J}}_x \,|_{\,\hat{x}_{k-1}} P_{k-1} \underline{\boldsymbol{J}}_x^T \,|_{\,\hat{x}_{k-1}} + \underline{\boldsymbol{Q}}_{k-1} \tag{53}$$

5.2.3.3. State Update Equation

Let $\boldsymbol{h}(x) \, \epsilon \, \mathbb{R}^m$ be the observation nonlinear vector function, then the measurement equation is given as $\boldsymbol{z}_k = \ \boldsymbol{h}(x_k) + \ \eta_k$. Since the state update equation for the EKF is the same as the KF, the state update equation becomes

$$\hat{x}_k = \hat{x}_{k(-)} + \underline{\boldsymbol{K}}_k (z_k - \boldsymbol{h}(x_k)) \tag{54}$$

5.2.3.4. Updated Error Covariance

Defining the error to be $e_k = \hat{x}_k - \ x_k$, and following a similar treatment of the error to find error covariance matrix for the KF yields,

$$\underline{\boldsymbol{P}}_k = \left(\boldsymbol{I} - \ \underline{\boldsymbol{K}}_k \, \underline{\boldsymbol{J}}_h \,|_{\,\hat{x}_{k(-)}}\right) \underline{\boldsymbol{P}}_{k(-)} \left(\boldsymbol{I} - \ \underline{\boldsymbol{K}}_k \, \underline{\boldsymbol{J}}_h \,|_{\,\hat{x}_{k(-)}}\right)^T + \underline{\boldsymbol{K}}_k \underline{\boldsymbol{R}}_k \underline{\boldsymbol{K}}_k^T \tag{55}$$

5.2.3.5. Kalman Gain

The Kalman Gain is found by minimizing the $Tr(\underline{\boldsymbol{P}}_k)$ with respect to $\underline{\boldsymbol{K}}_k$

$$\frac{\partial \left(Tr(\underline{\boldsymbol{P}}_k)\right)}{\partial \underline{\boldsymbol{K}}_k} = -\left(\underline{\boldsymbol{J}}_h \,|_{\,\hat{x}_{k(-)}} \underline{\boldsymbol{P}}_{k(-)}\right)^T - \underline{\boldsymbol{P}}_{k(-)} \underline{\boldsymbol{J}}_h \,|_{\,\hat{x}_{k(-)}} + 2\underline{\boldsymbol{K}}_k \, \underline{\boldsymbol{J}}_h \,|_{\,\hat{x}_{k(-)}} \underline{\boldsymbol{P}}_{k(-)} \, \underline{\boldsymbol{J}}_h^T \,|_{\,\hat{x}_{k(-)}} \underline{\boldsymbol{K}}_k^T + 2\underline{\boldsymbol{K}}_k \underline{\boldsymbol{R}}_k = 0 \tag{56}$$

where $\underline{\boldsymbol{J}}_h \big|_{\hat{x}_{k(-)}} \epsilon\ \mathbb{R}^{m\,x\,n}$ is the Jacobian of $\boldsymbol{h}(\cdot)$ about the point $\hat{x}_{k(-)}$. The resulting Kalman Gain is

$$\underline{\boldsymbol{K}}_k = \underline{\boldsymbol{P}}_{k(-)}\ \underline{\boldsymbol{J}}_h^T\big|_{\hat{x}_{k(-)}} \left(\underline{\boldsymbol{J}}_h\big|_{\hat{x}_{k(-)}} \underline{\boldsymbol{P}}_{k(-)}\ \underline{\boldsymbol{J}}_h^T\big|_{\hat{x}_{k(-)}} - \underline{\boldsymbol{R}}_k\right)^{-1} \tag{57}$$

Substituting the Kalman Gain into Equation (55)(57) yields a simplified form of the error covariance matrix

$$\underline{\boldsymbol{P}}_k = \left(\boldsymbol{I} - \underline{\boldsymbol{K}}_k\,\underline{\boldsymbol{J}}_h\big|_{\hat{x}_{k(-)}}\right) \underline{\boldsymbol{P}}_{k(-)} \tag{58}$$

5.3. *Relative O/D Filter Implementation*

5.3.1. Overview

Accurate relative orbit determination is essential for the successful completion of any proximity operations maneuver, whether it is completed via ground-in-the-loop or autonomous control. The latter option is critical if communications coverage is not continuous. As a result, an optimal sequential state estimator, such as a Kalman Filter, is desirable for autonomous proximity operations since it allows for real-time state estimation. More importantly, since the measurements are processed as they occur, corrective maneuvers can be performed as necessary. This is particularly of importance due to the varying range of uncertainties associated with the IPAs.

5.3.2. Linearized EKF Formulation

5.3.2.1. Initial State Estimate

In order to converge on a solution, any Kalman filter variant must have an initial state estimate to seed the filter. The algorithm to acquire the initial guess is detailed in Table 6. Through testing, an initial error covariance of $\underline{\boldsymbol{P}}_{k-1} = 10\,\boldsymbol{I}_{6\text{x}6}$ was found to work. The rationale for choosing a large initial covariance is that if the filter is working properly, the error covariance will shrink and converge to a ‘steady-state’ value.

Table 6. Relative O/D Initial Guess Algorithm

Relative O/D Initial Guess Algorithm

8. Compute an average relative position vector based on ten (10) images.

```
char ImageArray[639][479] = NULL;
char *ImgPtr;
ImgPtr = &ImageArray[0][0];
float InitialGuess[2];
for(char count = 0; count++; count < 10)
{
      ImgPtr = AcquireImage();
      InitialGuess += IPA_Routine();
}
InitialGuess = InitialGuess/10.;
```

9. Compute the components of the rotation matrix

$$\underline{\boldsymbol{R}}_d = [\widehat{\boldsymbol{R}} \quad \widehat{\boldsymbol{S}} \quad \widehat{\boldsymbol{W}}]^T$$

where

$$\widehat{\boldsymbol{R}} := \frac{\boldsymbol{r}_{chaser}}{||\boldsymbol{r}_{chaser}||}, \ \widehat{\boldsymbol{W}} := \frac{\boldsymbol{r}^x_{chaser}\boldsymbol{v}_{chaser}}{\left\|\boldsymbol{r}^x_{chaser}\boldsymbol{v}_{chaser}\right\|}, \text{ and } \widehat{\boldsymbol{S}} := \frac{\widehat{\boldsymbol{W}}^x\widehat{\boldsymbol{R}}}{\left\|\widehat{\boldsymbol{W}}^x\widehat{\boldsymbol{R}}\right\|}$$

10. Evaluate only the arguments of Equations (4)a, (5)a, (6)a, and (7)a:

Equation (4)a:

$$\eta_4 = \pm\frac{1}{2}\sqrt{Tr(\underline{\boldsymbol{R}}) + 1}$$

Equation (5)a:

$$\eta_1 = \pm\frac{1}{2}\sqrt{1 + \underline{\boldsymbol{R}}_{1,1} - \underline{\boldsymbol{R}}_{2,2} - \underline{\boldsymbol{R}}_{3,3}}$$

Equation (6)a:

$$\eta_2 = \pm\frac{1}{2}\sqrt{1 - \underline{\boldsymbol{R}}_{1,1} + \underline{\boldsymbol{R}}_{2,2} - \underline{\boldsymbol{R}}_{3,3}}$$

Equation (7)a:

$$\eta_3 = \pm\frac{1}{2}\sqrt{1 - \underline{\boldsymbol{R}}_{1,1} - \underline{\boldsymbol{R}}_{2,2} + \underline{\boldsymbol{R}}_{3,3}}$$

11. Determine the largest value between $\eta_1, \eta_2, \eta_3, \eta_4$

if $(\eta_1 > \{\eta_2, \eta_3, \eta_4\})$

- Use Equations (5)

else if $(\eta_2 > \{\eta_1, \eta_3, \eta_4\})$

- Use Equations (6)

else if $(\eta_3 > \{\eta_2, \eta_1, \eta_4\})$

- Use Equations (7)

else:

- Use Equations (4)

12. Compute $\boldsymbol{q}_d$ using the appropriate set of equations determined from Step 5.

13. Compute the rotation operator $\underline{\boldsymbol{Q}}(\boldsymbol{q}_e^{-1}) = \underline{\boldsymbol{Q}}\left(\begin{bmatrix}-\boldsymbol{\varepsilon}_e \\ \eta_e\end{bmatrix}\right)$

$$\underline{\boldsymbol{Q}}\left(\begin{bmatrix}-\boldsymbol{\varepsilon}_e \\ \eta_e\end{bmatrix}\right) = (\eta_e^2 - \boldsymbol{\varepsilon}_e^T\boldsymbol{\varepsilon}_e\ )\underline{\boldsymbol{I}} + 2(\boldsymbol{\varepsilon}_e\boldsymbol{\varepsilon}_e^T + \eta_e\boldsymbol{\varepsilon}_e^x)$$

(Note: This equation differs from Equation (13) by a single sign difference as it takes into account that the rotation operator is being computed for an inverse quaternion as opposed to a quaternion.)

14. Rotate *InitialGuess* into the RSW frame and set equal to $\widehat{\boldsymbol{x}}_{k-1}$

$$\widehat{\boldsymbol{x}}_{k-1} = \underline{\boldsymbol{Q}}(\boldsymbol{q}_e^{-1})\begin{bmatrix}X_B \\ Y_B \\ Z_B\end{bmatrix}$$

15. Initial Error Covariance

$$\underline{\boldsymbol{P}}_{k-1} = 10\,\boldsymbol{I}_{6x6}$$

5.3.2.2. State Prediction and Prediction Error Covariance

For the purposes of relative navigation, the state vector, $\boldsymbol{x} \epsilon \mathbb{R}^6$, is taken to be $\boldsymbol{x} = [x_{RSW}, y_{RSW}, z_{RSW}, \dot{x}_{RSW}, \dot{y}_{RSW}, \dot{z}_{RSW}]^T$. Furthermore, the use of the Clohessy-Wiltshire (CW) equations provides a unique solution to a set of second-order linear differential equations whose solution is given as the sum of the homogenous solution, $\underline{\boldsymbol{\Phi}}(t)$, and a particular solution, $\underline{\boldsymbol{\psi}}(t)$ [22]. For the sake of generality, the solution to a linear, time-varying, second order differential equation takes the form of

$$\boldsymbol{x}(t) = \underline{\boldsymbol{A}}(t, t_0)\boldsymbol{x}(t_0) + \int_{t_0}^{t} \underline{\boldsymbol{B}}(t, t_0)\, \boldsymbol{u}(\tau)\, d\tau \tag{59}$$

where $\underline{\boldsymbol{A}} \epsilon \mathbb{R}^{n \times n}$ is a time-varying, constant coefficient matrix, $\underline{\boldsymbol{B}} \epsilon \mathbb{R}^{n \times m}$ is the time-varying, constant coefficient input matrix, and $\boldsymbol{u}(\tau) \epsilon \mathbb{R}^m$ is the input vector.

The Prox-1 spacecraft utilizes a mono-propellant hydrazine thruster that is controlled via an off-the-shelf micro solenoid valve and produces approximately 1N of thrust [23]. As a result of valve actuator limitations of the thruster, thrusts of less than 1N cannot be realized. Therefore, the CW equations with a constant thruster input can be written as

$$\boldsymbol{x}(t) = \underline{\boldsymbol{\Phi}}(t_k, t_0)\boldsymbol{x}(t_0) + \underline{\boldsymbol{\psi}}(t_k, t_0)\boldsymbol{u}\, \mathrm{u}(t - t_0) \tag{60}$$

where $\mathrm{u}(t - t_0)$ is the Heaviside step function, $\underline{\boldsymbol{\Phi}} \epsilon \mathbb{R}^{6x6}$ is the state transition matrix (STM) given as

$$\boldsymbol{\Phi}(t_k, t_0) = \begin{bmatrix} 4 - 3\cos(nt) & 0 & 0 & \dfrac{\sin(nt)}{n} & \dfrac{2 - 2\cos(nt)}{n} & 0 \\ 6(\sin(nt) - nt) & 1 & 0 & \dfrac{2\cos(nt) - 2}{n} & \dfrac{4\sin(nt) - 3nt}{n} & 0 \\ 0 & 0 & \cos(nt) & 0 & 0 & \dfrac{\sin(nt)}{n} \\ 3n\sin(nt) & 0 & 0 & \cos(nt) & 2\sin(nt) & 0 \\ 6n(\cos(nt) - 1) & 0 & 0 & -2\sin(nt) & -3 + 4\cos(nt) & 0 \\ 0 & 0 & -n\sin(nt) & 0 & 0 & \cos(nt) \end{bmatrix} \tag{61}$$

where n is the mean motion of the RSO and is defined to be $n = \sqrt{\frac{\mu_\oplus}{a^3}}$ and $\underline{\boldsymbol{\psi}} \epsilon \mathbb{R}^{6x3}$ is the particular solution matrix and is given as

$$\boldsymbol{\Psi}(t_k, t_0) = \begin{bmatrix} \frac{1}{n^2} - \frac{\cos(nt)}{n^2} & \frac{2t}{n} - \frac{2\sin(n*t)}{n^2} & 0 \\ -\frac{2t}{n} + \frac{2\sin(nt)}{n^2} & \frac{4}{n^2} - \frac{3t^2}{2} - \frac{4\cos(nt)}{n^2} & 0 \\ 0 & 0 & \frac{1}{n^2} - \frac{\cos(nt)}{n^2} \\ \frac{\sin(nt)}{n} & \frac{2}{n} - \frac{2\cos(nt)}{n} & 0 \\ -\frac{2}{n} + \frac{2\cos(nt)}{n} & -3t + \frac{4\sin(nt)}{n} & 0 \\ 0 & 0 & \frac{\sin(nt)}{n} \end{bmatrix} \tag{62}$$

Since the Relative O/D process will *not* coincide with a thrust maneuver, for the particular solution can be ignored. However, it is important to note that the particular solution will be used for estimating the position of the RSO after a thrust maneuver is performed. Resultantly, the state prediction equation for the KF is

$$\hat{\boldsymbol{x}}_{k(-)} = \underline{\boldsymbol{\Phi}}(t_k, t_0)\boldsymbol{x}_{k-1} \tag{63}$$

The prediction error covariance is given as

$$\underline{\boldsymbol{P}}_{k(-)} = \underline{\boldsymbol{\Phi}}(t_k, t_0)\underline{\boldsymbol{P}}_{k-1}\underline{\boldsymbol{\Phi}}(t_k - t_0)^T + \underline{\boldsymbol{Q}}_{k-1} \tag{64}$$

where $\underline{\boldsymbol{Q}}_{k-1} = E\{w_{k-1}w_{k-1}^T\} \epsilon \mathbb{R}^{n\,x\,n}$ is the Process noise covariance matrix. The Process noise is defined as

$$\underline{\boldsymbol{Q}}_{\mathrm{k}} = \int_{t_0}^{t_k} \underline{\boldsymbol{\Phi}}(t_k, t_0)\underline{\boldsymbol{G}}\underline{\boldsymbol{Q}}\underline{\boldsymbol{G}}^T \underline{\boldsymbol{\Phi}}(t_k, t_0)^T d\tau \tag{65}$$

Discretizing Equation (65) and assuming $t_k - t_{k-1} \approx 0$, the STM becomes $\underline{\boldsymbol{\Phi}}(t_k, t_{k-1}) \approx \boldsymbol{I}$. The Process noise then becomes

$$\underline{\boldsymbol{Q}}_{\mathrm{k}} = \underline{\boldsymbol{G}}\underline{\boldsymbol{Q}}\underline{\boldsymbol{G}}^T \Delta t \tag{66}$$

Defining $\underline{\boldsymbol{G}}$ to be

$$\underline{\boldsymbol{G}} = \begin{bmatrix} \underline{\boldsymbol{0}}_{3x3} & \underline{\boldsymbol{0}}_{3x3} \\ \underline{\boldsymbol{0}}_{3x3} & \underline{\boldsymbol{I}}_{3x3} \end{bmatrix} \tag{67}$$

$\underline{\boldsymbol{Q}}_{\mathrm{k}}$ becomes

$$\underline{\boldsymbol{Q}}_{\mathrm{k}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{X_{RSW}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{Y_{RSW}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{Z_{RSW}}^2 \end{bmatrix} \Delta t \tag{68}$$

Specific to the Prox-1 Relative Orbital Determination process, the process noise was chosen to be $\sigma^2_{X_{RSW}} = \sigma^2_{Y_{RSW}} = \sigma^2_{Z_{RSW}} \sim N(0{,}1e-6)$ since it was found to be only dependent upong the sampling rate of the imager.

5.3.2.3. State Update and Error Covariance

The Kalman Gain is given as

$$\underline{\boldsymbol{K}}_k = \underline{\boldsymbol{P}}_{k(-)}\underline{\boldsymbol{H}}_k^T\left(\underline{\boldsymbol{H}}_k\underline{\boldsymbol{P}}_{k(-)}\underline{\boldsymbol{H}}_k^T - \underline{\boldsymbol{R}}_k\right)^{-1} \tag{69}$$

where $\underline{\boldsymbol{H}}_k \,\epsilon\, \mathbb{R}^{3\times 6}$ is the linear measurement model and is given as

$$\underline{\boldsymbol{H}}_k = [\underline{\boldsymbol{I}}_{3x3} \quad \underline{\boldsymbol{0}}_{3x3}] \tag{70}$$

since no operations are needed to be done on the measurements to map them to the state variables; and $\underline{\boldsymbol{R}}_k = E\{\eta_k\eta_k^T\} \,\epsilon\, \mathbb{R}^{m\times m}$ and is the measurement noise.

The measurement noise matrix is derived from the expected noise produced by the IPAs. Analysis of the IPAs has yielded that the noise is a function of range and is not constant. The resulting measurement noise matrix is given to be

$$\underline{\boldsymbol{R}}_k = \begin{bmatrix} e_i|\boldsymbol{z}_k|_{\ell^2} & 0 & 0 \\ 0 & e_i|\boldsymbol{z}_k|_{\ell^2} & 0 \\ 0 & 0 & e_i|\boldsymbol{z}_k|_{\ell^2} \end{bmatrix} \tag{71}$$

where $|\boldsymbol{z}_k|_{L_2}$is the Euclidean norm of the measurement vector and $e_i \sim N(0{,}1)$. It is important to note that $\underline{\boldsymbol{R}}_k$ is in the Body-Fixed frame and must be rotated into the RSW frame prior to integration into the KF. This is accomplished by

$$\underline{\boldsymbol{R}}_{k_{RSW}} = \underline{\boldsymbol{Q}}(\boldsymbol{q}_e^{-1})\underline{\boldsymbol{R}}_k\underline{\boldsymbol{Q}}(\boldsymbol{q}_e) \tag{72}$$

where $\underline{\boldsymbol{Q}}(\boldsymbol{q}_e^{-1})$ is given by Equation (13) For more details on the Body-Fixed to RSW coordinate frame transformation, see Section 3.3.2.2.

It is important to note that the measurement vector $\boldsymbol{z}_k$ is anchored in the body-fixed frame and must be mapped to the RSW reference frame before being applied to the KF. The resulting state update equation for the Relative O/D filter is given as

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k(-)} + \underline{\boldsymbol{K}}_k(\boldsymbol{z}_{k_{RSW}} - \underline{\boldsymbol{H}}_k\hat{\boldsymbol{x}}_{k(-)}) \tag{73}$$

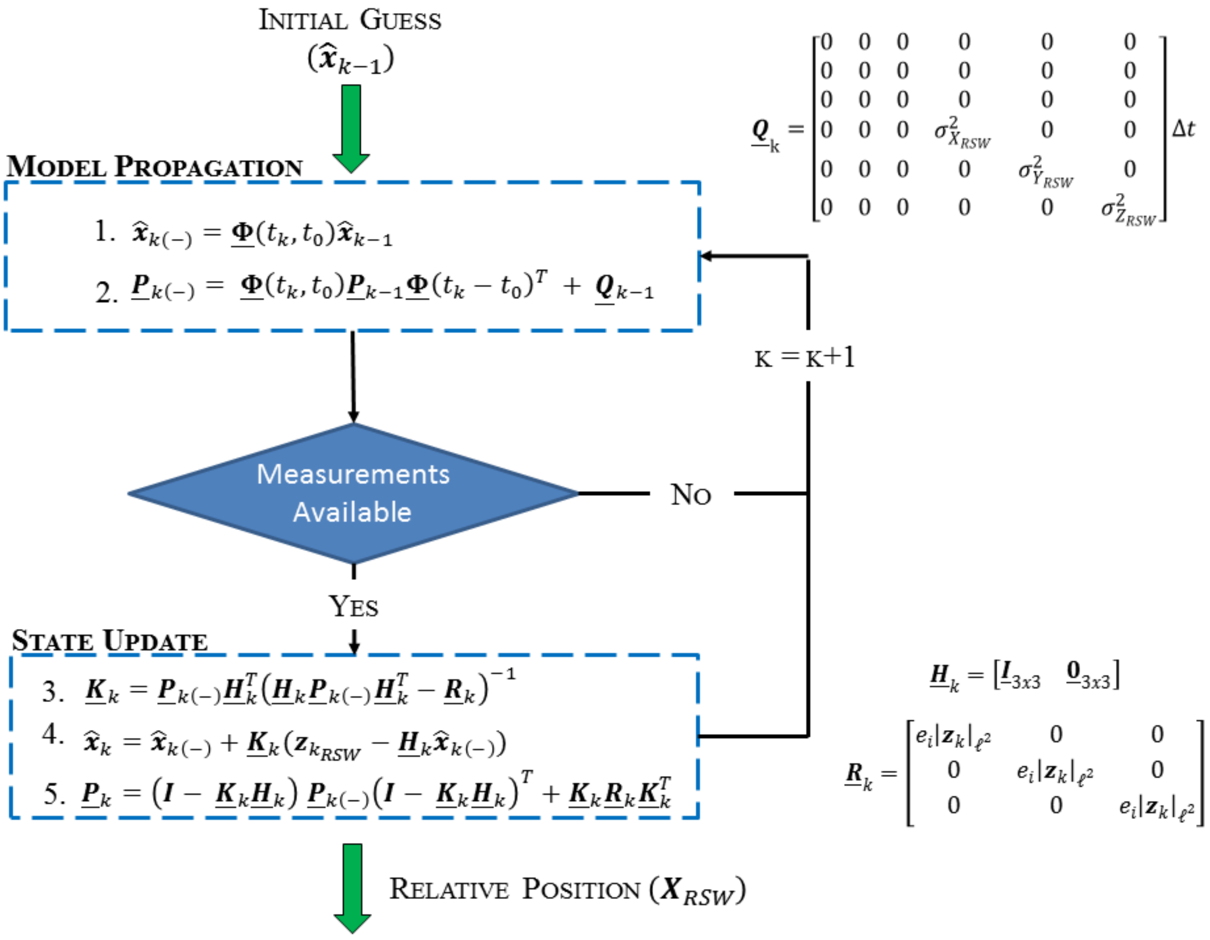
where $\underline{\boldsymbol{H}}_k \,\epsilon\, \mathbb{R}^{3\,x\,6}$ is the linear measurement model and $\boldsymbol{z}_{k_{RSW}} \,\epsilon\, \mathbb{R}^3$ is the measurement vector anchored in the RSW frame and is given as

$$\boldsymbol{z}_{k_{RSW}} = \underline{\boldsymbol{Q}}(\boldsymbol{q}_e^{-1})\boldsymbol{z}_k \tag{74}$$

Lastly, the update error covariance matrix is given as

$$\underline{\boldsymbol{P}}_k = \left(\boldsymbol{I} - \underline{\boldsymbol{K}}_k\underline{\boldsymbol{H}}_k\right)\underline{\boldsymbol{P}}_{k(-)}\left(\boldsymbol{I} - \underline{\boldsymbol{K}}_k\underline{\boldsymbol{H}}_k\right)^T + \underline{\boldsymbol{K}}_k\underline{\boldsymbol{R}}_k\underline{\boldsymbol{K}}_k^T \tag{75}$$

The algorithm for the Relative Orbital Determination Filter is illustrated in Figure 26. Given an initial guess, the filter propagates both an estimated state and error covariance. If measurements are available, the filter then corrects the predicted state estimate and updates the error covariance. It can be seen visually here that without measurements, the error covariance grows with respect to the process noise, $\underline{\boldsymbol{Q}}_{\mathrm{k}}$. Only when measurements are available can the filter begin to converge on a solution and reduce thee error covariance.



INITIAL GUESS
$(\hat{\boldsymbol{x}}_{k-1})$
MODEL PROPAGATION
1. $\hat{\boldsymbol{x}}_{k(-)} = \underline{\boldsymbol{\Phi}}(t_k,t_0)\hat{\boldsymbol{x}}_{k-1}$
2. $\underline{\boldsymbol{P}}_{k(-)} = \underline{\boldsymbol{\Phi}}(t_k,t_0)\underline{\boldsymbol{P}}_{k-1}\underline{\boldsymbol{\Phi}}(t_k-t_0)^T + \underline{\boldsymbol{Q}}_{k-1}$
$\underline{\boldsymbol{Q}}_k = \begin{bmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&\sigma^2_{X_{RSW}}&0&0\\0&0&0&0&\sigma^2_{Y_{RSW}}&0\\0&0&0&0&0&\sigma^2_{Z_{RSW}}\end{bmatrix}\Delta t$
K = K+1
Measurements Available
NO
YES
STATE UPDATE
3. $\underline{\boldsymbol{K}}_k = \underline{\boldsymbol{P}}_{k(-)}\underline{\boldsymbol{H}}_k^T\left(\underline{\boldsymbol{H}}_k\underline{\boldsymbol{P}}_{k(-)}\underline{\boldsymbol{H}}_k^T - \underline{\boldsymbol{R}}_k\right)^{-1}$
4. $\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k(-)} + \underline{\boldsymbol{K}}_k\left(\boldsymbol{z}_{k_{RSW}} - \underline{\boldsymbol{H}}_k\hat{\boldsymbol{x}}_{k(-)}\right)$
5. $\underline{\boldsymbol{P}}_k = \left(\boldsymbol{I} - \underline{\boldsymbol{K}}_k\underline{\boldsymbol{H}}_k\right)\underline{\boldsymbol{P}}_{k(-)}\left(\boldsymbol{I} - \underline{\boldsymbol{K}}_k\underline{\boldsymbol{H}}_k\right)^T + \underline{\boldsymbol{K}}_k\underline{\boldsymbol{R}}_k\underline{\boldsymbol{K}}_k^T$
$\underline{\boldsymbol{H}}_k = \left[\underline{\boldsymbol{I}}_{3x3} \quad \underline{\boldsymbol{0}}_{3x3}\right]$
$\underline{\boldsymbol{R}}_k = \begin{bmatrix}e_i|\boldsymbol{z}_k|_{\ell^2}&0&0\\0&e_i|\boldsymbol{z}_k|_{\ell^2}&0\\0&0&e_i|\boldsymbol{z}_k|_{\ell^2}\end{bmatrix}$
RELATIVE POSITION $(\boldsymbol{X}_{RSW})$

Figure 26. Relative Orbit Determination Filter Algorithm.

5.3.3. Relative O/D Filter Performance

To tune the Relative O/D navigation solution, a closed loop test of consisting of flight-like images were used to generate relative measurements via IPAs and fed into the KF to produce final relative position measurements in the RSW frame.

5.3.3.1. Truth State and Image Generation

Given an initial position and relative velocity, the CW equations were used to propagate a relative orbit for the designated length of the simulation. The resulting relative state vectors were then used to create a range measurement to be fed into the image generator developed by the Prox-1 AutoNav team. For the purposes of this test, perfect target acquisition (i.e. the target is always within the FOV of the imager) was assumed, although perfect pointing was not. LightSail was placed at a random locations within the FOV (however at the predetermined range) to simulate the effects of rotational drift and therefore put the relative orbit determination system through a much more thorough test. The initial orientation of LightSail was also randomized and its orientation rotated throughout the test to simulate moving around the target in a NMC.

Once the images were acquired, they were then processed using the IPA's and therefore produced a relative position vector in the body-fixed frame. For the purposes of this test, the truth data was then used to produce the necessary rotation to change the measurement to the RSW frame for use in the relative OD filter. This data was then fed through the Kalman filter and the results compared to the truth state.

5.3.3.2. Sample Scenario

In this scenario, a NMC is simulated with an initial position of -75m in the in-track direction and an initial radial velocity. The orbit is designed such that there is no motion in the cross-track direction. The Kalman filter is given an incorrect initial position with an incorrect in-track component as well as a small cross-track and radial components. The initial state vector given to the Kalman filter also has incorrect radial and cross-track velocity components. The simulation was then propagated for one orbit.

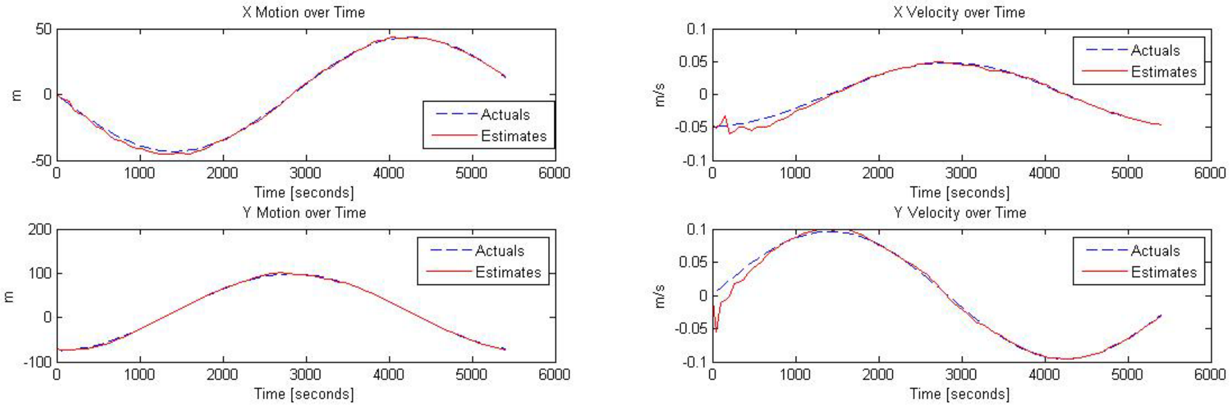


Figure 27. Accuracy of the Relative O/D Kalman Filter

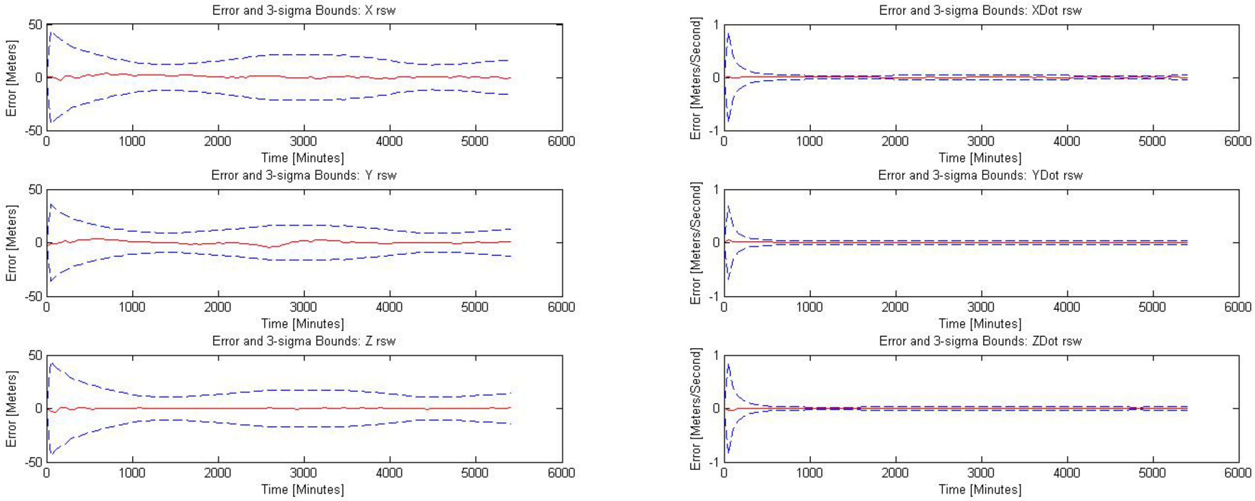


Figure 28. Relative O/D Error and covariance tracking

As illustrated by Figure 27 and Figure 28, the navigation filter rapidly tracks to the actual orbit and exhibits minimal error while the navigation solution is tracking the orbit. Table 7 lists the performance metrics of the Relative O/D filter.

Table 7. Relative O/D Filter Performance Metrics

Parameter	*Value*
Position Nominal Error	±5m
Position 3σ Error	±15m
Velocity Nominal Error	±0.02 m/s
Velocity 3σ Error	±0.10 m/s

5.4. *No-Measurement Navigation Strategy*

5.4.1. Relative O/D

5.4.1.1. State Estimate Propagator

During nominal proximity operations, constant relative position measurements will not be provided to the relative orbit determination filter for two reasons. First, image processing is computationally expensive; therefore there exists a finite sample rate (on the order of tens of seconds) between each image to account for the limitations imposed by the Prox-1 flight computer. Second, thrusting maneuvers will require slews that move RSO out of the microbolometer's FOV. However, although no relative position measurements are acquired during this period, Prox-1's GNC control logic will require a relative state estimate for maneuver planning and execution. It is for this reason that a state estimate propagator must be introduced into Prox-1's navigation system to continuously provide relative state estimates between periods of measurement.

The State Estimate Propagator will consist of two components; the steady state propagator and the force propagator. The steady state propagator will be used in all states where no measurement is being taken which include but are not limited to: time period between measurements, slew periods, and periods where RSO is temporarily lost or out of range. The force propagator will be used to update the state estimate when a known a thrust maneuver is applied. The updated state will aid in quick convergence of the Kalman filter once the measurement period begins again. Additionally, the propagated estimate will also aid in reacquisition of RSO after the slew and thrust maneuvers. Prox-1 tracking controllers will therefore be configured to slew to the expected location of RSO and not the previous location, minimizing any necessary search periods. If the RSO is not present, the propagated error covariance is then used to create an ellipsoidal search whose bounds are designated by the 3σ errors.

5.4.1.2. Steady State Propagation

Between measurements, the GNC control logic will require an updated relative position vector. The propagated relative state vector is determined using the homogenous solution to the CW Equations. For convenience (and not for taking up more space, I promise), Equation **Error! Reference source not found.** is repeated,

$$\hat{\boldsymbol{x}}_k = \underline{\boldsymbol{\Phi}}(t_k, t_{k-1})\hat{\boldsymbol{x}}_{k-1} \tag{76}$$

where $n = \sqrt{\frac{\mu_\oplus}{a^3}}$ and

$$\boldsymbol{\Phi}(t_k, t_0) = \begin{bmatrix} 4 - 3\cos(nt) & 0 & 0 & \dfrac{\sin(nt)}{n} & \dfrac{2 - 2\cos(nt)}{n} & 0 \\ 6(\sin(nt) - nt) & 1 & 0 & \dfrac{2\cos(nt) - 2}{n} & \dfrac{4\sin(nt) - 3nt}{n} & 0 \\ 0 & 0 & \cos(nt) & 0 & 0 & \dfrac{\sin(nt)}{n} \\ 3n\sin(nt) & 0 & 0 & \cos(nt) & 2\sin(nt) & 0 \\ 6n(\cos(nt) - 1) & 0 & 0 & -2\sin(nt) & -3 + 4\cos(nt) & 0 \\ 0 & 0 & -n\sin(nt) & 0 & 0 & \cos(nt) \end{bmatrix} \tag{77}$$

and the error covariance is given by Equation **Error! Reference source not found.**. As mentioned previously, since no measurements are being made, the error covariance grows with respect to the process noise.

5.4.1.3. Force Propagator

Whenever a thrust maneuver is performed, thee resulting force applied to the Prox-1 bus is accounted for in the state estimation by utilizing both the homogenous and particular solutions to the CW equations. The discretized equivalent to Equation **Error! Reference source not found.** is

$$\widehat{\boldsymbol{x}}_{k(-)} = \underline{\boldsymbol{\Phi}}(t_k, t_{k-1})\widehat{\boldsymbol{x}}_{k-1} + \underline{\boldsymbol{\psi}}(t_k, t_{k-1})\boldsymbol{u} \tag{78}$$

where $\boldsymbol{u} \epsilon \mathbb{R}^3$ is the thruster input $n = \sqrt{\frac{\mu_\oplus}{a^3}}$ and

$$\boldsymbol{\Psi}(t_k, t_0) = \begin{bmatrix} \dfrac{1}{n^2} - \dfrac{\cos(nt)}{n^2} & \dfrac{2t}{n} - \dfrac{2\sin(n * t)}{n^2} & 0 \\ -\dfrac{2t}{n} + \dfrac{2\sin(nt)}{n^2} & \dfrac{4}{n^2} - \dfrac{3t^2}{2} - \dfrac{4\cos(nt)}{n^2} & 0 \\ 0 & 0 & \dfrac{1}{n^2} - \dfrac{\cos(nt)}{n^2} \\ \dfrac{\sin(nt)}{n} & \dfrac{2}{n} - \dfrac{2\cos(nt)}{n} & 0 \\ -\dfrac{2}{n} + \dfrac{2\cos(nt)}{n} & -3t + \dfrac{4\sin(nt)}{n} & 0 \\ 0 & 0 & \dfrac{\sin(nt)}{n} \end{bmatrix} \tag{79}$$

6. Controller Formulation

6.1. *Slew and Tracking Controller*

6.1.1. Overview

Spacecraft attitude control methods range can from low accuracy, passive control techniques such as gravity-gradient boom and passive magnetic control, to more accurate, active control mechanisms such as control moment gyros (CMGs), reaction wheels, and thrusters. However, attitude control via angular momentum exchange devices (AMED) is more desirable given the fact that their life-span is not dictated by the amount of propellant stored on-board and can achieve a higher degree of precision than an attitude thruster. However, control algorithms involving an AMED require more detailed knowledge of the dynamics and physical limitations of the AMED is needed so that unrealistic actuator commands are not instructed by the control laws.

6.1.2. Error Quantity Definitions

For the sake of clarity, two (2) error quantities, the error quaternion, $\boldsymbol{q}_e$, and the error spacecraft angular rate, $\boldsymbol{\omega}_e$, are respectively defined as:

$$\boldsymbol{q}_e = \left[\, \underline{\Xi}(\boldsymbol{q}_d^{-1})\ \boldsymbol{q}_d^{-1}\, \right]\boldsymbol{q} \tag{80}$$

$$\boldsymbol{\omega}_e = \boldsymbol{\omega}_B - \underline{\boldsymbol{R}}_e\, \boldsymbol{\omega}_d \tag{81}$$

where $\boldsymbol{q} = \begin{bmatrix} \boldsymbol{\varepsilon} \\ \eta \end{bmatrix}$, and $\boldsymbol{\varepsilon} \in \mathbb{R}^{3x1}$ and $\eta \in \mathbb{R}^1$, $\underline{\boldsymbol{R}}_e \in \mathbb{R}^{3x3}$ is the rotation matrix that maps a vector in $\mathbb{R}^{3x3}$ from the desired reference frame to the body-fixed reference frame and is defined as

$$\underline{\boldsymbol{R}}_e = \underline{\Xi}^T(\boldsymbol{q}_e)\underline{\boldsymbol{\Psi}}(\boldsymbol{q}_e) \tag{82}$$

where $\underline{\Xi}(\boldsymbol{q})$: $\mathbb{R}^4 \to \mathbb{R}^{4x3}$ and is defined as

$$\underline{\Xi}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{\varepsilon}^x + \eta\underline{\mathbf{I}} \\ -\boldsymbol{\varepsilon}^T \end{bmatrix} \tag{83}$$

and $\underline{\boldsymbol{\Psi}}(\boldsymbol{q})$: $\mathbb{R}^4 \to \mathbb{R}^{4x3}$ and is defined as

$$\underline{\boldsymbol{\Psi}}(\boldsymbol{q}) = \begin{bmatrix} -\boldsymbol{\varepsilon}^x + \eta\underline{\mathbf{I}} \\ -\boldsymbol{\varepsilon}^T \end{bmatrix} \tag{84}$$

The error attitude kinematics can be defined by using Equation (12) and Equation (81)

$$\dot{\boldsymbol{q}}_e = \frac{1}{2}\underline{\Xi}(\boldsymbol{q}_e)\boldsymbol{\omega}_e \tag{85}$$

which can be equivalently written as

$$\dot{\boldsymbol{q}}_e = \frac{1}{2}\underline{\boldsymbol{\Omega}}(\boldsymbol{\omega}_e)\boldsymbol{q}_e \tag{86}$$

where $\underline{\boldsymbol{\Omega}} \epsilon \mathbb{R}^{4x4}$ and is defined as

$$\underline{\boldsymbol{\Omega}}(\boldsymbol{\omega}) = \begin{bmatrix} -\boldsymbol{\omega}^{x} & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{T} & 0 \end{bmatrix} \tag{87}$$

6.1.3. Slew and Tracking Controller (STC) Formulation

In order to realize the appropriate torque commands that will drive both the vector and scalar portion of the error quaternion zero (0) and unity respectively as well as null the error angular rate, the following Lyapunov candidate function (LCF) is considered

$$V = \frac{1}{2}\boldsymbol{\omega}_e^T\underline{\boldsymbol{K}}^{-1}\underline{\boldsymbol{J}}\boldsymbol{\omega}_e + \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} + (1-\eta^2) \tag{88}$$

where $\underline{\boldsymbol{K}} \epsilon \mathbb{R}^{3x3}$ is a small, positive-definite gain matrix and $\underline{\boldsymbol{J}} \epsilon \mathbb{R}^{3x3}$ is the centroidal inertia matrix without the AMED. In order for the candidate function considered in Equation (88) to be valid, it must obey the following criteria:

$$\begin{aligned} &1.\ V(\boldsymbol{\omega}_e, \boldsymbol{q}, t) > 0 \quad && \forall \quad && \boldsymbol{\omega}_e, \boldsymbol{q} \neq \boldsymbol{0} \\ &\quad\ V(\boldsymbol{\omega}_e, \boldsymbol{q}, t) = 0 && for && \boldsymbol{\omega}_e, \boldsymbol{q} = 0 \\ &2.\ \dot{V}(\boldsymbol{\omega}_e, \dot{\boldsymbol{\omega}}_e, \boldsymbol{q}, t) < 0 && \forall && \boldsymbol{\omega}_e, \dot{\boldsymbol{\omega}}_e, \boldsymbol{q} \to \boldsymbol{0} \end{aligned} \tag{89}$$

By satisfying these criteria, the resulting control input will be asymptotically stable, in the sense of Lyapunov. Asymptotic stability is defined as follows [24].

Definition 1:

An equilibrium point $\hat{\boldsymbol{x}}$ is asymptotically stable, in the sense of Lyapunov, if:

i. $\forall\, \varepsilon > 0,\ \exists\, \delta > 0$ such that if $\left|\left|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}\right|\right| < \delta$ then $\left|\left|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}\right|\right| < \varepsilon\ \forall\, t \geq 0$

and

ii. $\exists\, \delta > 0$ such that if $\left|\left|\boldsymbol{x}_0 - \hat{\boldsymbol{x}}\right|\right| < \delta$ then $\lim_{t\to\infty} \boldsymbol{x}(t) = \hat{\boldsymbol{x}}$

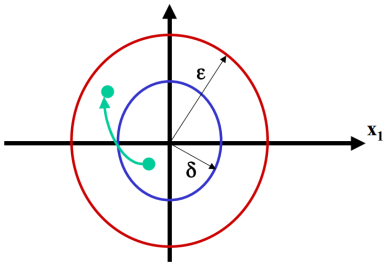


Figure 29. Lyapunov stability diagram.

From the quaternion unity constraint, the identity $\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = 1 - \eta^2$ can be substituted into and simplifying Equation (88)

$$V = \frac{1}{2}\boldsymbol{\omega}_e^T\underline{\boldsymbol{K}}^{-1}\underline{\boldsymbol{J}}\boldsymbol{\omega}_e + 2\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} \tag{90}$$

Differentiating the LCF,

$$\dot{V} = \frac{1}{2}\left(\dot{\boldsymbol{\omega}}_e^T\underline{\boldsymbol{K}}^{-1}\underline{\boldsymbol{J}}\boldsymbol{\omega}_{\boldsymbol{e}} + \boldsymbol{\omega}_e^T\underline{\boldsymbol{K}}^{-1}\underline{\boldsymbol{J}}\dot{\boldsymbol{\omega}}_{\boldsymbol{e}}\right) + 2(\dot{\boldsymbol{\varepsilon}}^T\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^T\dot{\boldsymbol{\varepsilon}}) \tag{91}$$

Since $\boldsymbol{a}^T\dot{\boldsymbol{a}} = \dot{\boldsymbol{a}}^T\boldsymbol{a}$, Equation (91) can be simplified to

$$\dot{V} = \boldsymbol{\omega}_e^T\underline{\boldsymbol{K}}^{-1}\underline{\boldsymbol{J}}\dot{\boldsymbol{\omega}}_{\boldsymbol{e}} + \boldsymbol{4}\boldsymbol{\varepsilon}^T\dot{\boldsymbol{\varepsilon}} \tag{92}$$

Substituting the vector component of the time rate of change of the error attitude, $\dot{\boldsymbol{\varepsilon}} = \frac{1}{2}(\boldsymbol{\omega}_e^x\boldsymbol{\varepsilon} + \eta_e\boldsymbol{\omega}_e)$ yields

$$\dot{V} = \boldsymbol{\omega}_e^T\underline{\boldsymbol{K}}^{-1}\underline{\boldsymbol{J}}\dot{\boldsymbol{\omega}}_{\boldsymbol{e}} + \boldsymbol{4}\boldsymbol{\varepsilon}^T\left(\frac{1}{2}(\boldsymbol{\omega}_e^x\boldsymbol{\varepsilon} + \eta_e\boldsymbol{\omega}_e)\right) \tag{93}$$

Differentiating Equation (81) prior to substituting into Equation (93),

$$\dot{\boldsymbol{\omega}}_e = \dot{\boldsymbol{\omega}}_B - \dot{\underline{\boldsymbol{R}}}_e\boldsymbol{\omega}_d - \underline{\boldsymbol{R}}_e\dot{\boldsymbol{\omega}}_d \tag{94}$$

yields

$$\dot{V} = \boldsymbol{\omega}_e^T\underline{\boldsymbol{K}}^{-1}\left(\underline{\boldsymbol{J}}\dot{\boldsymbol{\omega}}_B - \underline{\boldsymbol{J}}\dot{\underline{\boldsymbol{R}}}_e\boldsymbol{\omega}_d - \underline{\boldsymbol{J}}\underline{\boldsymbol{R}}_e\dot{\boldsymbol{\omega}}_d\right) + \boldsymbol{4}\boldsymbol{\varepsilon}^T\left(\frac{1}{2}(\boldsymbol{\omega}_e^x\boldsymbol{\varepsilon} + \eta_e\boldsymbol{\omega}_e)\right) \tag{95}$$

where

$$\dot{\underline{\boldsymbol{R}}}_e = \underline{\boldsymbol{\Xi}}^T(\dot{\boldsymbol{q}}_e)\underline{\boldsymbol{\Psi}}(\boldsymbol{q}_e) + \underline{\boldsymbol{\Xi}}^T(\boldsymbol{q}_e)\underline{\boldsymbol{\Psi}}(\dot{\boldsymbol{q}}_e) \tag{96}$$

Performing the matrix operations, the time rate-of-change of the error attitude rotation matrix simplifies to

$$\dot{\underline{\boldsymbol{R}}}_e = -\boldsymbol{\omega}_e^x\underline{\boldsymbol{R}}_e \tag{97}$$

Substituting Equation (97) into Equation (95) yields

$$\dot{V} = \boldsymbol{\omega}_e^T\underline{\boldsymbol{K}}^{-1}\left(\underline{\boldsymbol{J}}\dot{\boldsymbol{\omega}}_B + \underline{\boldsymbol{J}}\boldsymbol{\omega}_e^x\underline{\boldsymbol{R}}_e\boldsymbol{\omega}_d - \underline{\boldsymbol{J}}\underline{\boldsymbol{R}}_e\dot{\boldsymbol{\omega}}_d\right) + 2\boldsymbol{\varepsilon}^T(\boldsymbol{\omega}_e^x\boldsymbol{\varepsilon} + \eta_e\boldsymbol{\omega}_e) \tag{98}$$

By using the following identities,

$$\begin{aligned} \boldsymbol{a}^T\boldsymbol{b} &= \boldsymbol{b}^T\boldsymbol{a} && (a) \\ \boldsymbol{a}^x\boldsymbol{b} &= -\boldsymbol{b}^x\boldsymbol{a} && (b) \\ (\boldsymbol{a}^x)^{\boldsymbol{T}}\boldsymbol{b} &= -(\boldsymbol{b}^x)^T\boldsymbol{a} && (c) \end{aligned} \tag{99}$$

the second half of Equation (98) becomes

$$\begin{aligned} 2\boldsymbol{\varepsilon}^T(\boldsymbol{\omega}_e^x\boldsymbol{\varepsilon} + \eta_e\boldsymbol{\omega}_e) &= 2\boldsymbol{\omega}_e^T(-\boldsymbol{\varepsilon}_e^x\boldsymbol{\varepsilon}_e + \eta_e\boldsymbol{\varepsilon}_e) \\ &= 2\eta_e\boldsymbol{\omega}_e^T\boldsymbol{\varepsilon}_e \end{aligned} \tag{100}$$

since the cross-product of a vector and itself is zero (0). Substituting back into Equation (98) and simplifying

$$\dot{V} = \boldsymbol{\omega}_e^T\underline{\boldsymbol{K}}^{-1}\left(\underline{\boldsymbol{J}}\dot{\boldsymbol{\omega}}_B + \underline{\boldsymbol{J}}\boldsymbol{\omega}_e^x\underline{\boldsymbol{R}}_e\boldsymbol{\omega}_d - \underline{\boldsymbol{J}}\underline{\boldsymbol{R}}_e\dot{\boldsymbol{\omega}}_d + 2\eta_e\underline{\boldsymbol{K}}\boldsymbol{\varepsilon}_e\right) \tag{101}$$

From Euler's Second Law,

$$\underline{\boldsymbol{J}}\dot{\boldsymbol{\omega}}_B = \boldsymbol{\tau} - \boldsymbol{\omega}_B^x \underline{\boldsymbol{J}}\boldsymbol{\omega}_B \tag{102}$$

where $\boldsymbol{\tau} \in \mathbb{R}^3$ is the control torque applied on the spacecraft by an AMED in the body-fixed reference frame. Substituting Equation (102) into Equation (101)

$$\dot{V} = \boldsymbol{\omega}_e^T \underline{\boldsymbol{K}}^{-1} \left(\boldsymbol{\tau} - \boldsymbol{\omega}_B^x \underline{\boldsymbol{J}}\boldsymbol{\omega}_B + \underline{\boldsymbol{J}}\boldsymbol{\omega}_e^x \underline{\boldsymbol{R}}_e \boldsymbol{\omega}_d - \underline{\boldsymbol{J}}\underline{\boldsymbol{R}}_e \dot{\boldsymbol{\omega}}_d + 2\eta_e \underline{\boldsymbol{K}}\boldsymbol{\varepsilon}_e \right) \tag{103}$$

In order for this LCF to meet the second criteria listed in Equation (89), $\boldsymbol{\tau}$ is chosen to be

$$\boldsymbol{\tau} = \boldsymbol{\omega}_B^x \underline{\boldsymbol{J}}\boldsymbol{\omega}_B - \underline{\boldsymbol{J}}\boldsymbol{\omega}_e^x \underline{\boldsymbol{R}}_e \boldsymbol{\omega}_d + \underline{\boldsymbol{J}}\underline{\boldsymbol{R}}_e \dot{\boldsymbol{\omega}}_d - 2\eta_e \underline{\boldsymbol{K}}\boldsymbol{\varepsilon}_e - \underline{\boldsymbol{C}}\boldsymbol{\omega}_e \tag{104}$$

where $\underline{\boldsymbol{C}} \in \mathbb{R}^{3x3}$ is small, positive-definite gain matrix. Substituting the control torque into Equation (103) yields

$$\dot{V} = -\boldsymbol{\omega}_e^T \underline{\boldsymbol{K}}^{-1} \underline{\boldsymbol{C}}\boldsymbol{\omega}_e \tag{105}$$

Resultantly, the time-derivative of the LCF is in-fact negative definite and is a valid LCF as long as $\underline{\boldsymbol{K}}^{-1}\underline{\boldsymbol{C}}$ is positive-definite. In order to ensure this condition, one such selection of $\underline{\boldsymbol{K}}$ and $\underline{\boldsymbol{C}}$ is $\underline{\boldsymbol{K}} = k\underline{\boldsymbol{J}}$ and $\underline{\boldsymbol{C}} = c\underline{\boldsymbol{J}}$ where $k \in \mathbb{R}^1$ and $c \in \mathbb{R}^1$ are small and positive [16].

To insure only realizable actuator commands are requested by the STC, a saturation function is employed and is defined as follows.

Definition 2:

A saturation function of an n-dimensional vector $\boldsymbol{v} = [v_1, v_2, \dots, v_n]^T$ is defined as

$$sat(\boldsymbol{v}) = \begin{bmatrix} sat_1(v_1) \\ sat_2(v_2) \\ \vdots \\ sat_n(v_n) \end{bmatrix}$$

where

$$sat_i(v_i) = \begin{cases} v_i^+ & \text{if } \mathrm{v_i} > \mathrm{v_i^+} \\ v_i & \text{if } \mathrm{v_i^-} \le \mathrm{v_i} \le \mathrm{v_i^+} \\ v_i^- & \text{if } \mathrm{v_i} < \mathrm{v_i^-} \end{cases} \tag{106}$$

Applying Definition 2 to the STC, the resulting controller input $\boldsymbol{u}$ is given as

$$\boldsymbol{u} = sat(\boldsymbol{\tau}) \tag{107}$$

where

$$sat(\boldsymbol{\tau}) = \begin{cases} \gamma \tau_{max}^+ & \text{if } \tau_\mathrm{i} > \tau_{\mathrm{i}_{max}}^+ \\ \tau_i & \text{if } \tau_{\mathrm{i}_{max}}^- \le \tau_\mathrm{i} \le \tau_{\mathrm{i}_{max}}^+ \\ \gamma \tau_{max}^- & \text{if } \tau_\mathrm{i} < \tau_{\mathrm{i}_{max}}^- \end{cases} \tag{108}$$

$\gamma \epsilon (0,1]$ is a scaling parameter, $\tau_{i\,max}^{+}$ is the maximum positive toque about the i-th axis, $\tau_{i\,max}^{-}$ is the maximum negative torque about the i-th axis, and the control torque is

$$\boldsymbol{\tau} = \boldsymbol{\omega}_B^x \underline{\boldsymbol{J}} \boldsymbol{\omega}_B - \underline{\boldsymbol{J}} \boldsymbol{\omega}_e^x \underline{\boldsymbol{R}}_e \boldsymbol{\omega}_d + \underline{\boldsymbol{J}} \underline{\boldsymbol{R}}_e \dot{\boldsymbol{\omega}}_d - 2\eta_e \underline{\boldsymbol{K}} \boldsymbol{\varepsilon}_e - \underline{\boldsymbol{C}} \boldsymbol{\omega}_e \qquad (109)$$

Typically, for most AMEDs, $\tau_{i\,max}^{+} = \tau_{i\,max}^{-}$.

6.1.4. STC Development & Validation

6.1.4.1. STC Development Environment

A generic spacecraft AMED controller block diagram is illustrated in Figure 30. Typically, the AMED controller is developed without detailed *a priori* knowledge of any physical limitations of the control actuators or mechanisms, such as angular momentum saturation limits, while more basic information, such as torque limitations, are known.

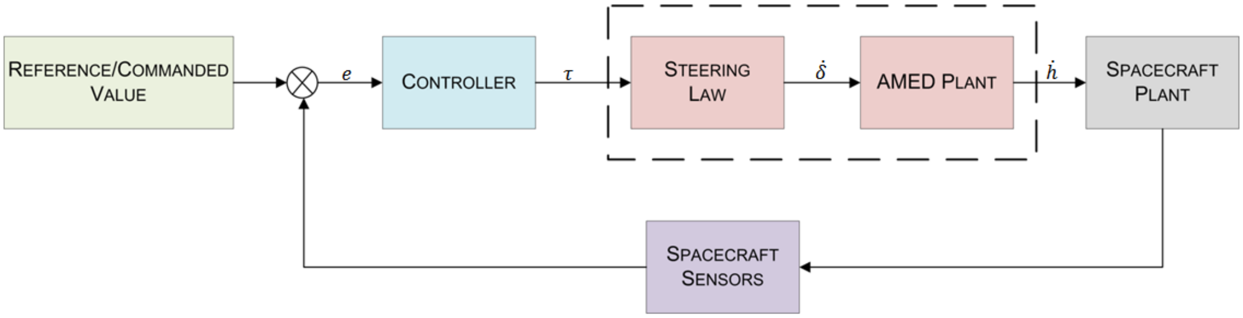


Figure 30. Generic attitude control block diagram.

Figure 31 illustrates the top-level initial testing environment for the STC. It is important to note that this environment does not simulate the Angular Momentum Exchange Devices (AMEDs) used by Prox-1 - specifically, Control Moment Gyros (CMGs). The spacecraft properties used in this testing environment are similar to the current best estimates (CBEs) for the Prox-1 spacecraft.

Additionally, in the environment shown in Figure 31, only the slew portion of the STC is being tested – as the variables pertaining to tracking a desired reference frame – specifically $\underline{\boldsymbol{R}}_e$, $\dot{\underline{\boldsymbol{R}}}_e$, and $\boldsymbol{\omega}_d$ – contain inputs of zero (0). In order to test the tracking capabilities of the STC, information a more sophisticated testing environment is needed as information regarding the desired frame must be inputted into the controller. However, since the basic functionality of a tracking controller relies on the functionality of the slew component, the tracking controller component is secondary in importance.

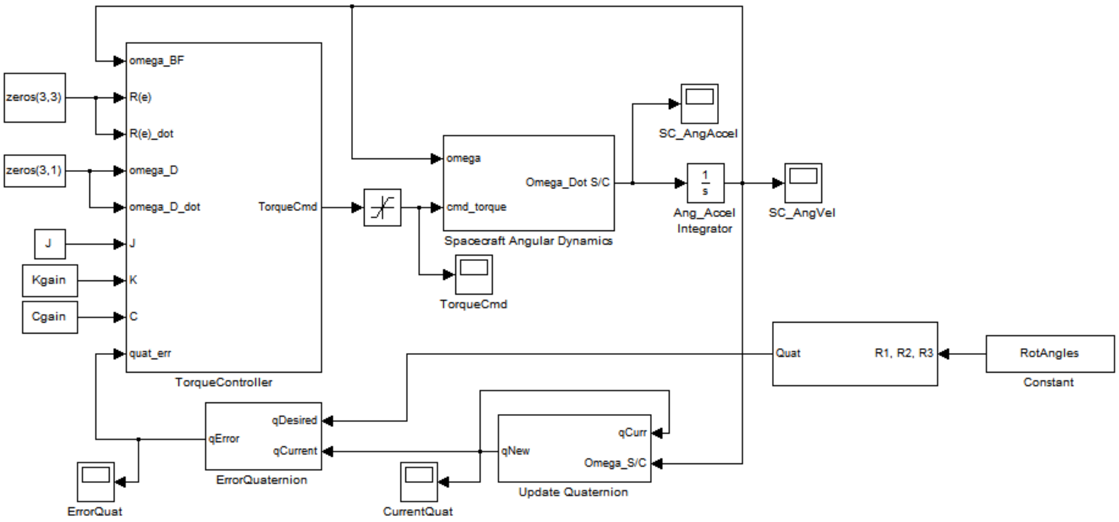


Figure 31. Top-level STC testing environment.

6.1.4.2. Controller Gain Selection

In order to properly select the gains, a range of gains was considered for both $k \epsilon (0, 1.5]$ and $c \epsilon (0\ 1.5]$ along with the following metrics for a 90º-180º-270º slew about the spacecraft *x, y,* and *z* axes respectively:

1. AMED control input limitations 2. Slew Completion Time 3. Error Angle

In order to avoid both CMG saturation and hardware anomalies, an upper control torque limit of 75% of $\tau_{\max}$ was selected. Additionally, the slew convergence criterion of 1x10-5 radians was selected for the maximum error angle allowable.

The resulting plot comparing the gains of c and k to the slew completion time is shown in Figure 32. The minimum slew completion time that meets the constraints listed above

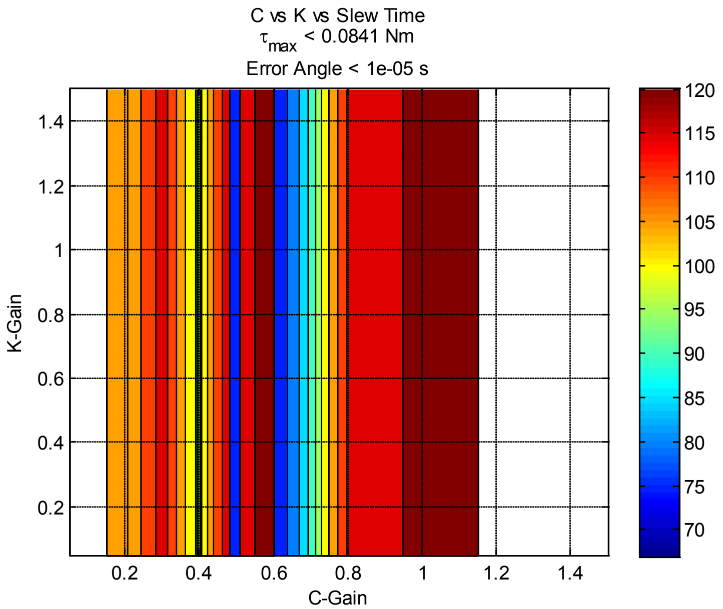


Figure 32. Comparison of gains c and k as a function of slew compeltion time.

is approximately 70 seconds. In order to minimize the control torque input – and therefore minimizing power consumption – the closed-loop system must be critically damped - that is, ξ is unity. By imposing this constraint and realizing that the slew completion time is invariant under *k*, the approximation for the value of *k* can be rewritten in terms of the scalar value *c*, $k \approx \frac{1}{4}c^2$. Additionally, the range of values for the scalar gain *c* that meet the constraints imposed and minimize the slew completion time is [0.485 , 0.505]; the corresponding range of optimal values for the scalar gain *k* is [0.0588 0.0638]. However, it is important to note that since the control torque input is non-linear, these ranges are not definite. Resultantly, these ranges were opened up by 20% in each direction to account for the non-linearity of the control torque input.

The resulting gains from the analysis for the scalar gains *c* and *k* are listed in Table 8.

Table 8. Selected gains for the STC.

Parameter	*Value*
k	0.05
c	0.5

6.1.5. STC Performance

Given the gains in Table 8 , the resulting control torque input is illustrated in Figure 33.

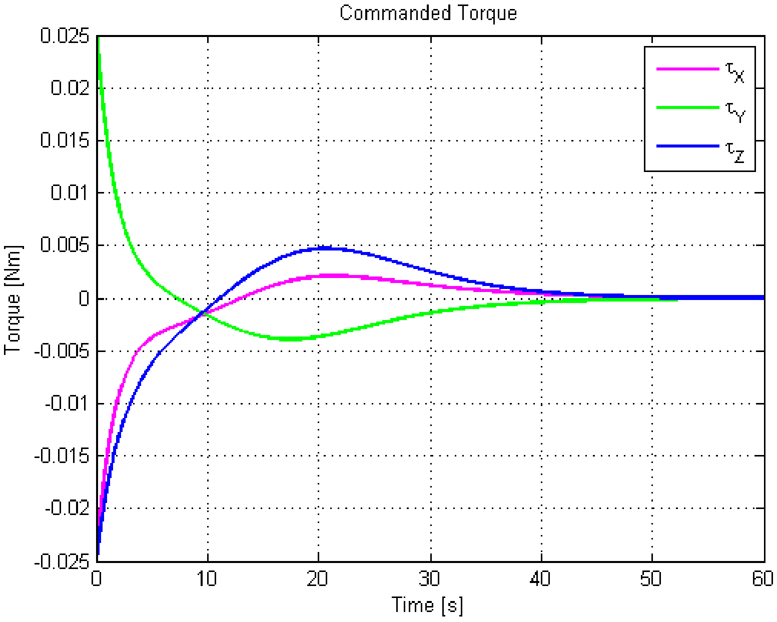
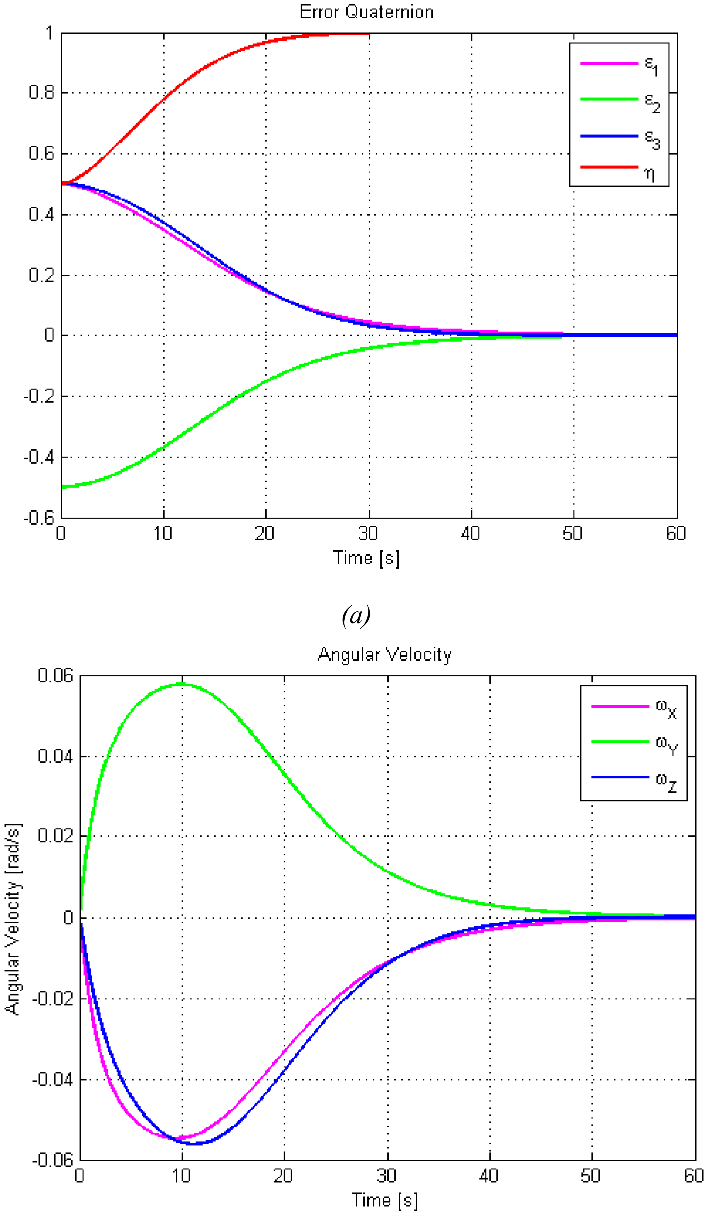


Figure 33. STC control torque input

Other performance parameters, such as the error quaternion, spacecraft angular rates, spacecraft angular acceleration, and inertial quaternion are illustrated in Figure 34. As illustrated by Figure 34(a), the error quaternion demonstrates that the selected gains do in-fact produce a critically damped system.



(a)

(c)

Figure 34. STC performance parameters: (a) error quaternion; (b) spacecraft angular acceleration; (c) inertial quaternion; (d) spacecraft angular rate.

6.1.6. Convergence Criteria

Convergence of the STC is measured using both the error angle and the norm of the error angular rate. The error angle is defined as

$$\theta_e = \cos^{-1} \eta_e \tag{110}$$

As the spacecraft converges on the desired quaternion, the vector component of the error quaternion goes to zero (0); as a result of the quaternion unity constraint, the scalar component of the error quaternion goes to unity consequently causing the error angle to go to zero(0). Expressing the error angle in terms of the vector component of the error quaternion, Equation (110) becomes

$$\theta_e = \cos^{-1} \sqrt{1 - \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon}} \tag{111}$$

The value for θ_e is dependent upon the precision of the Inertial Attitude Determination (A/D) filter.

The second part of the STC convergence criteria is the norm of the error angular rates. As defined by Equation (81), the error angular rate is a combination of both the spacecraft angular rate, $\boldsymbol{\omega}_B$, and the rotation of the desired frame, $\boldsymbol{\omega}_d$**,** with respect to the body frame. In other words, for a rest-to-rest maneuver, $\boldsymbol{\omega}_d$ (as defined by Equation (81)) is zero (0) and the error angular rates is purely a measure of the spacecraft angular rate; for a tracking maneuver, the error angular rate is a comparison of the spacecraft angular compared to the angular rate of the desired frame. Therefore, the norm of the error angular rate is a valid metric for convergence. It is important to note here that the value signaling convergence of the angular rate is dependent upon the resolution of the rate gyros in the inertial measurement unit (IMU). The resolution of typical rate gyros is dependent upon the dynamic sensitivity range selected as the gain used to amplify the signal is larger for larger dynamic ranges.

6.2. *De-tumble Controller*

6.2.1. Overview

Once released from a launch vehicle, a spacecraft will typically begin to tumble at some angular rates proportional to the moment applied on the spacecraft by the separation event divided by the inertia of the spacecraft. For most satellites, the random tumbling produced by the separation event creates an undesirable environment in which the spacecraft cannot successfully complete its mission. As a result, these angular rates must be damped prior to satellite checkout and commissioning. In the case of the Prox-1 mission, the de-tumble maneuver will be performed via magnetic torque rods. Compared to other attitude controlling mechanisms, magnetic torque rods trade the mechanical complexity and higher torque capabilities for a more reliable actuator that relies solely on renewable electricity at lower power levels. This is a particularly important consideration since, after launch, the satellite's batteries may not be fully charged. As a result, the time of the launch and the state of the batteries at separation can have a significant impact on when the satellite will not only startup but the length of time between the separation event and a stable attitude being achieved.

6.2.2. De-Tumble Controller (DTC) Formulation

Like the development of the STC, the control input for the de-tumble controller (DTC) will be derived using the following LCF,

$$V = \frac{1}{2}\boldsymbol{\omega}_e^T \underline{\boldsymbol{K}}^{-1} \underline{\boldsymbol{J}} \boldsymbol{\omega}_{\boldsymbol{e}} \tag{112}$$

where $\boldsymbol{\omega}_e \epsilon \mathbb{R}^3$ is the error angular velocity, $\underline{\boldsymbol{K}} \epsilon \mathbb{R}^{3x3}$ is a small, positive-definite gain matrix and $\underline{\boldsymbol{J}} \epsilon \mathbb{R}^{3x3}$ is the centroidal inertia matrix. Since the DTC is anchored only in the BFCF, it is worth noting the error angular velocity is simply the angular velocity in the body frame, $\boldsymbol{\omega}_B$. In order for the candidate function considered in Equation (112) to be valid, it must obey the following criteria:

$$\begin{array}{lccc} 1.\ V(\boldsymbol{\omega}_B, t) > 0 & \forall & \boldsymbol{\omega}_B \neq \boldsymbol{0} \\ \quad V(\boldsymbol{\omega}_B, t) = 0 & for & \boldsymbol{\omega}_B = 0 \\ 2\ \ \dot{V}(\boldsymbol{\omega}_B, t) < 0 & \forall & \boldsymbol{\omega}_B \to \boldsymbol{0} \end{array} \tag{113}$$

Differentiating the LCF,

$$\dot{V} = \frac{1}{2}\left(\dot{\boldsymbol{\omega}}_B^T \underline{\boldsymbol{K}}^{-1} \underline{\boldsymbol{J}} \boldsymbol{\omega}_B + \boldsymbol{\omega}_B^T \underline{\boldsymbol{K}}^{-1} \underline{\boldsymbol{J}} \dot{\boldsymbol{\omega}}_B\right) \tag{114}$$

Knowing $\boldsymbol{a}^T \dot{\boldsymbol{a}} = \dot{\boldsymbol{a}}^T \boldsymbol{a}$, Equation (114) becomes

$$\dot{V} = \boldsymbol{\omega}_B^T \underline{\boldsymbol{K}}^{-1} \underline{\boldsymbol{J}} \dot{\boldsymbol{\omega}}_B \tag{115}$$

Substituting Equation (102) yields

$$\dot{V} = \boldsymbol{\omega}_B^T \underline{\boldsymbol{K}}^{-1} (\boldsymbol{\tau} - \boldsymbol{\omega}_B^x \underline{\boldsymbol{J}} \boldsymbol{\omega}_B) \tag{116}$$

In order Equation (116) to be negative definite, the control torque is chosen to be

$$\boldsymbol{\tau} = \boldsymbol{\omega}_B^x \underline{\boldsymbol{J}} \boldsymbol{\omega}_B - \left(\underline{\boldsymbol{C}} - \widehat{\boldsymbol{b}} \widehat{\boldsymbol{b}}^T\right) \boldsymbol{\omega}_B \tag{117}$$

where $\widehat{\boldsymbol{b}} = \frac{\boldsymbol{b}}{|\boldsymbol{b}|} \epsilon \mathbb{R}^3$ is the unit vector parallel to the local geomagnetic vector expressed in the body-fixed frame and $\underline{\boldsymbol{C}} \epsilon \mathbb{R}^{3x3}$ is a positive definite gain matrix [25]. Substituting the control torque into Equation (116) yields

$$\dot{V} = \boldsymbol{\omega}_B^T \underline{\boldsymbol{K}}^{-1} \left(\boldsymbol{\omega}_B^x \underline{\boldsymbol{J}} \boldsymbol{\omega}_B - \left(\underline{\boldsymbol{C}} - \widehat{\boldsymbol{b}} \widehat{\boldsymbol{b}}^T\right) \boldsymbol{\omega}_B - \boldsymbol{\omega}_B^x \underline{\boldsymbol{J}} \boldsymbol{\omega}_B\right) \tag{118}$$

which simplifies to

$$\dot{V} = -\boldsymbol{\omega}_B^T \underline{\boldsymbol{K}}^{-1}\left(\underline{\boldsymbol{C}} - \widehat{\boldsymbol{b}}\widehat{\boldsymbol{b}}^T\right)\boldsymbol{\omega}_B \tag{119}$$

It is easily seen that the $\lim_{\boldsymbol{\omega}_B \to 0} \dot{V} = 0$ and that $\dot{V}$ vanishes whenever $\boldsymbol{\omega}_B$ is parallel to the local geomagnetic field.

However, if possible, this situation needs to be avoided as when the spacecraft rotates about an axis parallel to the local magnetic field vector, the (magnetic) torque rods will not be able to apply a torque in the axis of rotation since

$$\boldsymbol{\tau} = \boldsymbol{m}^x \boldsymbol{b} \tag{120}$$

where $\boldsymbol{m} \epsilon \mathbb{R}^3$ is the commanded magnetic moment of the spacecraft. By introducing an orthogonality constraint, $\boldsymbol{m}^T \widehat{\boldsymbol{b}} = 0$, Equation (120) becomes

$$\boldsymbol{m} = \frac{1}{|\boldsymbol{b}|}\widehat{\boldsymbol{b}}^x \boldsymbol{\tau} \tag{121}$$

and combing with the control torque specified in Equation (117), the control input $\boldsymbol{u} \epsilon \mathbb{R}^3$ is

$$\boldsymbol{u} = -\frac{1}{|\boldsymbol{b}|}\widehat{\boldsymbol{b}}^x \left(\boldsymbol{\omega}_B^x \underline{\boldsymbol{J}}\boldsymbol{\omega}_B - \left(\underline{\boldsymbol{C}} - \widehat{\boldsymbol{b}}\widehat{\boldsymbol{b}}^T\right)\boldsymbol{\omega}_B\right) \tag{122}$$

The recommend gain selection for $\underline{\boldsymbol{C}}$ is given as

$$\underline{\boldsymbol{C}} = 2\pi\sqrt{\frac{\mu}{a}}\underline{\boldsymbol{J}} \tag{123}$$

7. Future Work

7.1. *Guidance*

The main area of improvement for the IPAs is in the area of orientation estimation of the RSO. From consultation of others, this author strongly feels that the incorporation of edge detection algorithms can help improve the accuracy of the determining the orientation of the RSO. However, due to time constraints, the author was not able to explore this opportunity further.

7.2. *Navigation*

The largest risk item in the Navigation section is the inertial attitude determination (A/D) filter. Without this filter, the orientation of the spacecraft with respect to the inertial frame cannot be determined. Due to time and resource constraints, the author was not able to actively pursue the completion of this filter personally.

7.3. *Control*

7.3.1. STC Future Work

The slew controller is completed and has been shown to work as an independent controller. The next step is integration into the automated GN&C system.

7.3.2. DTC Future Work

While the control input for the DTC has been derived, it has not been successfully tested due to simulation environment errors. The errors in the simulation environment are attributed in the following areas and will be addressed in the next revision of this document: errors in the magnetometer model as well as lack of controller bandwidth filtering.

7.4. *Automated GN&C System*

While the underlying framework has been created for automated proximity operations, the task of fully integrating all of the individual components that comprise the GN&C system into a generic logic structure that will allow for successful completion of both the rest-to-rest in-track maneuver as well as successful completion of NMC maneuvers around a RSO still remains. This can be accomplished in the MATLAB Simulink environment via inclusion of StateFlow blocksets which will aid with the various mode transitions to compelte aiding with mode transitions

Appendix A: Overview of Relative Motion

Introduction

Motion of a spacecraft can be characterized in several frames of references that can aide in conceptual understanding or aide in maneuver planning and orbit propagation. For the purpose of proximity operations between two spacecraft, the relative motion of one spacecraft with respect to the other is of particular importance. Rather than a traditional pseudo-inertial reference frame such as the Earth-Centered-Inertial (ECI) frame that describes motion of a spacecraft about the Earth, it is more beneficial to describe the spacecraft's motion in a non-inertial frame centered upon the other spacecraft.

The problem of relative motion was first studied by George W. Hill in 1886 to account for the motion of the lunar perigee. [26]Later Clohessy and Wiltshire (CW) formulated a similar approach specifically for spacecraft. [27]In both cases a series of linearized, time-invariant equations of motions were developed that described relative orbital motion.

Basic Equations of Motion

The formulation of the equations of motion for Prox-1 will follow Vallado's derivation[28], a similar derivation to Clohessy-Wiltshire with a few exceptions. First, define a frame centered upon a spacecraft, heretofore referred to as the 'Chief'. This frame will be defined using the RSW coordinate system:

- $\hat{R}$: Radial component, collinear with the position vector.
- $\hat{S}$: In-track component, in the direction of the Chief's velocity vector.
- $\widehat{W}$: Cross-track component, normal to the orbital plane, or $\hat{R}x\hat{S}$

This frame is also referred to as the Local-Vertical, Local-Horizontal (LVLH) frame. Within RSW, the relative position can be defined as $(x\hat{R}, y\hat{S}, z\widehat{W})$ and the relative velocity as $(\dot{x}\hat{R}, \dot{y}\hat{S}, \dot{z}\widehat{W})$.

Next, define the relative motion of a second spacecraft, called the "Chaser", with respect to the Chief.

$$\ddot{\vec{r}}_{rel} = \ddot{\vec{r}}_{Chaser} - \ddot{\vec{r}}_{Chief}$$

Additionally, define the angular rate of the Chief spacecraft's motion; this is equal to the mean motion of the orbit, as it is assumed that the Chief is in a circular orbit.

$$\omega = n = \sqrt{\frac{\mu}{r_{Chief}^3}}$$

The relative motion can be transformed into the RSW frame using a series of rotations and coordinate transformations. See Vallado for the full derivation. The results yield:

$$\ddot{\vec{r}}_{rel_R} = -\omega^2\left(x\hat{R} + y\hat{S} + z\widehat{W} - 3x\hat{R}\right) + \hat{F}_{Thrust} + 2\omega\dot{y}\hat{R} - 2\omega\dot{x}\hat{S} + \omega^2 x\hat{R} + \omega^2 y\hat{S}$$

Separating into each coordinate, the basic CW equations form a set of second-order differential equations:

$$\ddot{x} - 2\omega\dot{y} - 3\omega^2 x = f_x$$
$$\ddot{y} + 2\omega\dot{x} = f_y$$
$$\ddot{z} + \omega^2 z = f_z$$

Next, the unforced solution (F_{Thrust} = 0) can be solved, assuming near-circular motion. Using Laplace transforms for the ODEs, the following equations can be used to describe the relative motion of the Chaser:

$$x_u = \frac{\dot{x}_0}{n}\sin(nt) - \left(3x_0 + \frac{2\dot{y}_0}{n}\right)\cos(nt) + \left(4x_0 + \frac{2\dot{y}_0}{n}\right)$$
$$y_u = \frac{2\dot{x}_0}{n}\cos(\mathrm{nt}) + \left(6x_0 + \frac{4\dot{y}_0}{n}\right)\sin(nt) - (6nx_0 + 3\dot{y}_0)t - \frac{2\dot{x}_0}{n} + y_0$$
$$z_u = \frac{\dot{z}_0}{n}\sin(nt) + z_0\cos(nt)$$
$$\dot{x_u} = \frac{\dot{x}_0}{n}\cos(\mathrm{nt}) + (3nx_0 + 2\dot{y}_0)\sin(nt)$$
$$\dot{y}_u = -2\dot{x}_0\sin(\mathrm{nt}) + (6nx_0 + 4\dot{y}_0)\cos(\mathrm{nt}) - (6nx_0 + 3\dot{y}_0)$$
$$\dot{z}_u = \dot{z}_0\sin(nt) - nz_0\sin(nt)$$

Thrusting Equations of Motion

The forced equations of motion can also be solved, given a constant thrust magnitude and direction. In solving the set of differential equations, the equations of motion are a combination of the homogenous unforced solution and the particular solution. These equations can be seen below, where f_x refers to the acceleration due to thrusting in the radial (x) direction. [29]

$$x_f = x_u + \frac{f_x}{n^2}(1 - \cos(nt)) + 2\frac{f_y}{n}\left(t - \frac{\sin(nt)}{n}\right)$$
$$y_f = y_u + 4\frac{A_y}{n^2}(1 - \cos(nt)) - 2\frac{f_x}{n}\left(t - \frac{\sin(nt)}{n}\right) - \frac{3}{2}f_y t^2$$
$$z_f = z_u + 4\frac{f_z}{n^2}(1 - \cos(nt))$$
$$\dot{x}_f = \dot{x}_u + \frac{f_x}{n}\sin(nt) + 2\frac{f_y}{n}(1 - \cos(nt))$$
$$\dot{y}_f = \dot{y}_u + 4\frac{f_y}{n}\sin(nt) - 2\frac{f_x}{n}(1 - \cos(nt)) - 3\,f_y t$$

$$\dot{z}_f = \dot{z}_u + \frac{f_z}{n}\sin(nt)$$

Appendix B: Relative Orbital Elements

The relative motion of the chaser spacecraft may also be described by six parameters, analogous to the orbital elements used to describe inertial movement of a satellite. These parameters will be called Relative Orbital Elements (ROE). The equations for the ROEs can be seen below.

$$a_e = 2\sqrt{\left(\frac{\dot{x}}{n}\right)^2 + \left(3x + 2\frac{\dot{y}}{n}\right)^2}$$

$$x_d = 4x + \frac{2\,\dot{y}}{n}$$

$$y_d = y - 2\frac{\dot{x}}{n}$$

$$\beta = atan2(\dot{x}, 3nx + 2\dot{y})$$

$$z_{max} = \sqrt{\left(\frac{\dot{z}}{n}\right)^2 + z^2}$$

$$\gamma = atan2(nz, \dot{z}) - \beta$$

A diagram is useful to define a_e, x_d, y_d, and β; this can be seen in Figure 35. a_e defines the semi-major axis of the 2x1 ellipse of the chaser. The coordinates (x_d, y_d) define the location of the center of the ellipse. β^* defines the angle from perigee for the current position of the satellite in the X-Y plane, defined from perigee. Z_{max} defines the amplitude of the sinusoidal cross-track motion, and γ defines the angle between the X-Y plane and the relative orbital plane, as measured from the relative ascending node of the motion.

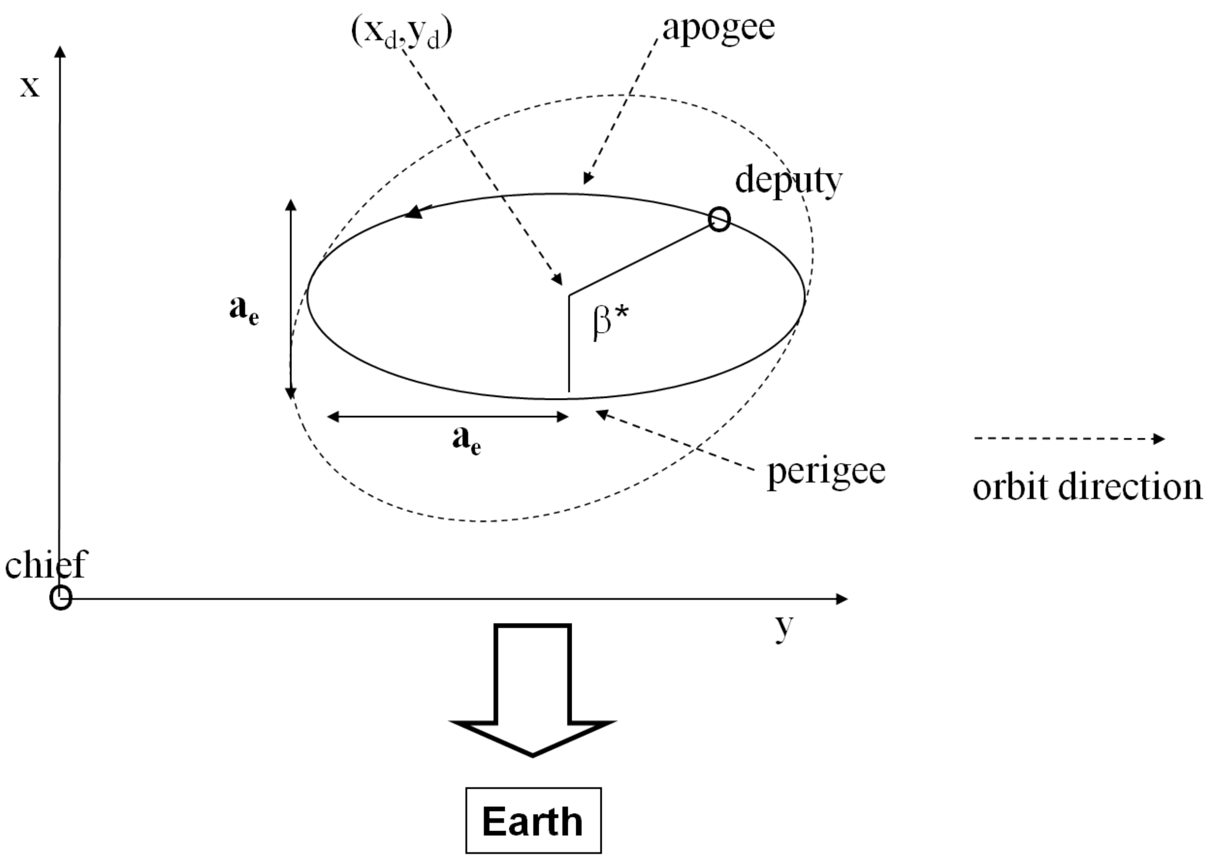


Figure 35 ROE Diagram (Credit: Lovell)

In unforced motion of the chaser spacecraft, the ROEs follow a linear set of equations, as seen below. Only y_d and β change in time for unforced motion; the other ROEs remain constant.

$$a_e = a_{e0}$$
$$x_d = x_{d0}$$
$$y_d = y_{d0} - \frac{3}{2} n x_{d0} t$$
$$\beta = \beta_0 + nt$$
$$z_{max} = z_{max0}$$
$$\gamma = \gamma_0$$

ROEs in Forced Motion

Using the thrusting EOMs described earlier, it is possible to re-parameterize them using ROEs to determine the effects of thrusting upon ROEs. These equations can be seen below; Δt refers to the burn time for continuous thrust maneuver, while t_f refers to the wait time from a particular state before a thrust maneuver begins.

$$a_e^+ = \sqrt{\begin{array}{c}\left[a_{e0}\sin\left(\beta_0 + n\left(t_{f1} + \Delta t\right)\right) + 2\frac{f_x}{n^2}\sin(n\Delta t) + 4\frac{f_y}{n^2}(1 - \cos(n\Delta t))\right]^2 \\ + \left[a_{e0}\cos\left(\beta_0 + n\left(t_{f1} + \Delta t\right)\right) + 4\frac{f_y}{n^2}\sin(n\Delta t) - 2\frac{f_x}{n^2}(1 - \cos(n\Delta t))\right]^2\end{array}}$$

$$x_d^+ = x_{d0} + 2\frac{f_y}{n}\Delta t$$

$$y_d^+ = y_{d0} - \frac{3}{2}nx_{d0}\left(t_{f1} + \Delta t\right) - \left(2\frac{f_x}{n} + \frac{3}{2}f_y\Delta t\right)\Delta t$$

$$\begin{aligned}\beta^+ = atan2\Bigg[\Bigg(&a_{e0}n\sin\left(\beta_0 + n\left(t_{f1} + \Delta t\right)\right) + 2\frac{f_x}{n}\sin(n\Delta t) \\ &+ 4\frac{f_y}{n}(1 - \cos(n\Delta t))\Bigg), \Bigg(a_{e0}n\cos\left(\beta_0 + n\left(t_{f1} + \Delta t\right)\right) + 2\frac{f_x}{n}\sin(n\Delta t) \\ &+ 4\frac{f_y}{n}(1 - \cos(n\Delta t))\Bigg)\Bigg]\end{aligned}$$

$$z_{max}^+ = \sqrt{\begin{array}{c}\left(z_{max0}\cos\left(\gamma_0 + \beta_0 + n\left(t_{f1} + \Delta t\right)\right) + \frac{f_x}{n^2}\sin(n\Delta t)\right)^2 + \\ \left(z_{max0}\sin\left(\gamma_0 + \beta_0 + n\left(t_{f1} + \Delta t\right)\right) + \frac{f_x}{n^2}(1 - \cos(n\Delta t))\right)^2\end{array}}$$

$$\gamma^+ = atan2\left[\left(z_{max0}n\sin\left(\gamma_0 + \beta_0 + n\left(t_{f1} + \Delta t\right)\right) + \tfrac{f_z}{n}(1 - \cos(n\Delta t))\right), \left(z_{max0}n\cos\left(\gamma_0 + \beta_0 + n\left(t_{f1} + \Delta t\right)\right) + \tfrac{f_z}{n}\sin(n\Delta t)\right)\right]$$

References

[1] D. A. Mindell, *Digital Apollo: Human and Macine in Spaceflight*, First MIT Press paperback Edition, 2011 ed. Cambridge, MA: The MIT Press, 2008.

[2] E. M. Hinman and D. Bushman, "Soviet Automated Rendezvous and Docking System Overview," *NASA Automated Rendezvous and Capture Review: Executive Summary,* 1991.

[3] I. Kawano, M. Mokuno, T. Kasai, and T. Suzuki, "Result of Autonomous Rendezvous Docking Experiment of Engineering Test Satellite-VII"," *Journal of Spacecraft and Rockets,* vol. 38, pp. 105-111, Jan - Feb 2001 2001.

[4] Y. Ohkami and I. Kawano, "Autonomous Rendezvous and Docking by Engineering Test Satellite VII; A challenge of Japan in Guidance, Navigation, and Control-Br," *Acta Astronautica,* vol. 53, pp. 1-8, July 2003 2003.

[5] AFRL. *XSS-11 Micro Satellite*. Available: http://www.vs.afrl.af.mil/FactSheets/XSS11-MicroSatellite.pdf

[6] M. A. Dornheim. (2006) Orbital Express to Test Full Autonomy for On-Orbit Service. *Aviation Week.*

[7] H. NASA Marshall Space Flight Center, Alabama, "DART Demonstrator to Test Future autonomous Rendezvous Technologies in Orbit," ed, 2004.

[8] J. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*: Princeton University Press, 1999.

[9] R. R. Bate, D. D. Mueller, and J. E. White, *Fundamentals of Astrodynamics*. New York, NY: Dover Publications, 1970.

[10] G. Xu, *Orbits*. Berlin: Springer, 2008.

[11] K. Jewett and K. DeLuca, "Technical Memorandum P1-TM-2013-019: Coordinate Frame Defintiions," Georgia Institutue of Technology, Center for Space Systems, Internal Technical Memorandum2013.

[12] J. R. Wertz, *Spacecraft Attitude Determination and Control*. Dordrecht, Holland: Reidel Publishing Company.

[13] M. D. Shuster, "Survey of Attitude Representations," *Journal of the Astronautical Science s,* vol. 41, pp. 439-517, 1993.

[14] "IEEE Standard for Floating-Point Arithmetic, *IEEE Std 754-2008*," ed: The Institute of Electrical and Electronics Engineers, Inc. , 2008.

[15] "IEEE Standard for Floating-Point Arithmetic, *IEEE Std 754-1985*," ed: The Institute of Electrical and Electronics Engineers, Inc. , 1985.

[16] H. Schildt, *The Complete Reference C*, Fourth ed.: McGraw-Hill Companies 2000.

[17] E. Bellet and D. A. Spencer, "Detection and Localization of a Target on a Thermal Image Using a "Blobber" Algorithm," Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA2011.

[18] N. Vedie and A. Srivastava, "Title," unpublished|.

[19] S. Sternberg, *Dynamical Systems*. Mineloa, NY: Dover Publications, Inc., 2010.

[20] N. Martinson, J. D. Munoz, and G. J. Wiens, "A new method of guidnace control for autonomous rendezvous in a cluttered space environment," presented at the AIAA Guidance, Navigation, and Control Conference, Hilton Head, SC, 2007.

[21] I. Lopez and C. R. McInnes, "Autonomous Rendezvous Using Aritifical Potential Function Guidance," *AIAA Guidance, Control, and Dynamics,* vol. 18, pp. 237-241, 1995.

[22] W. H. Clohessy and R. S. Wiltshire, "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciencens,* vol. 27, pp. 653-658, 1960.

[23] C. Biddy and T. Svitek, "Monopropellant Micropropulsion System for CubeSats," presented at the 23rd Annual AIAA/USU Confernce on Small Satellites, Logan, UT, 2009.

[24] SpaceQuest, "Technical Specifications: GPS-12-V1 Receiver Performance," ed, p. 3.

[25] G. Avanzini and F. Giulietti, "Magnetic Detumbling of a Rigid Spacecraft," *Journal of Guidance, Control, and Dynamics,* vol. 35, pp. 1326-1337, 2012.

[26] G. W. Hill, "Researches in Lunar Theory," *American Journal of Mathematics,* pp. 5-26, 1878.

[27] W. H. Clohessy and R. S. Wiltshire, "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences,* pp. 653-674, 1960.

[28] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*. Hawthorne, CA: Microcosm Press, 2007.

[29] H. Vasavada, "GN&C for Relative Motion via Continuous Thrusting," Air Force Research Laboratory Space Vehicles Directorate, Kirtland Air Force Base, NM2012.