The Pennsylvania State University

The Graduate School

SIMULTANEOUS POSE AND MAPPING OF A NON-COOPERATIVE MANEUVERING TARGET USING AN OCTREE-BASED APPROACH

A Dissertation in

Aerospace Engineering

by

Peter C Scarcella

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2021

The dissertation of Peter C Scarcella was reviewed and approved by the following:

David B. Spencer
Professor of Aerospace Engineering
Dissertation Advisor
Chair of Committee

Robert Melton
Professor of Aerospace Engineering

Puneet Singla
Professor of Aerospace Engineering

Minghui Zhu
Associate Professor of Electrical Engineering

Amy Pritchett
Professor of Aerospace Engineering
Head of Aerospace Engineering

ABSTRACT

Autonomous proximity operation missions in the space domain are a multi-faceted, studied problem. Performing such a mission around an uncooperative, unknown target presents many difficulties in estimating the relative pose of the target. If the target executes a controlled or uncontrolled (outgassing from a damaged satellite) low-thrust continuous maneuver, tracking or rendezvousing with the target becomes challenging and potentially dangerous especially in close proximity. A solution to this problem is desired such that the developed algorithm is amenable to onboard implementation, utilizes stereo vision, can estimate relative pose without initial relative information to the target, and can detect and estimate a continuous low-thrust maneuver.

The developed algorithm presented in this dissertation utilizes an octree based approach to construct a three-dimensional map of the target while simultaneously estimating the pose in a multiplicative extended Kalman filter. A coarse volume estimate and probabilistic uncertainty mapping are extrapolated from the octree map and used as the initialization of a consider parameter in a consider variable state dimension (VSD) filter following maneuver detection. This framework switches to a thrusting model after the Mahalanobis distance passes a pre-defined threshold initializing the consider parameters and starting an interacting multiple model (IMM) filter. Assuming an evenly-distributed density, eight density models comprising the models in the IMM establish a mechanism for thrust states to be estimated by combining the best amalgamation of density models and the coarse volume estimate. Essentially, an online process is created to indirectly obtain mass information from the octree map without directly estimating it. Since target volume is not completely observable, it is not directly estimated and instead used as a consider parameter with an associated covariance obtained from the octree mapping uncertainty.

Despite using coarse volume estimates, the simulated results show good convergence for pose and thrust states as well as accurate maneuver detection for continuous low-thrust maneuvers. Two thrust magnitudes were analyzed (differing by an order of magnitude) varying the moment of maneuver detection such that the octree mapping was at two different levels of completion. Results obtained from a complete octree mapping demonstrated accuracy both in relative pose estimates and maneuver detection. Convergence and maneuver detection had more prevalent errors associated with a less robust octree mapping as expected though their results still showed promise.

The developed algorithms create an online process of indirect mass estimation to enable the thrust estimation once a maneuver is detected. Presented results demonstrate the algorithm’s ability to detect unknown maneuvers and accurately estimate relative pose and thrust states. It is clearly shown that the octree map provides a clear effect on reducing the covariance and improving estimates of the pose and thrust. Furthermore, utilizing the consider filter, VSD, and IMM solely in the maneuvering model keeps the computational cost low.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Acronyms

BRDF. .... Bidirectional Reflectance Distribution Function

CoM. .... Center of Mass

CoG. .... Center of Geometry

CRP. .... Classical Rodrigues Parameters

CRSV. .... Compact Reference Satellite Variables

ECI. .... Earth Centered Inertial frame

EKF. .... Extended Kalman Filter

GPU. .... Graphical Processing Unit

IEKF .... Iterated Extended Kalman Filter

IMM. .... Interacting Multiple Model

iSAM .... Incremental Smoothing and Mapping

KLT .... Kanade Lucas Tomosi

LVLH. .... Local Vertical Local Horizontal

MAP. .... Maximum a Posteriori

MEKF. .... Multiplicative Extended Kalman Filter

MRP. .... Modified Rodrigues Parameters

NMC. .... Natural Motion Circumnavigation

OLAE. .... Optimal Linear Attitude Estimator

PCA. .... Principal Component Analysis

RANSAC. .... Random Sample and Consensus

RMS. .... Root Mean Square

RTS. .... Rauch-Tung-Striebel Smoother

SLAM. .... Simultaneous Localization and Mapping

SEKF.. Schmidt Extended Kalman Filter

STL. .... Stereolithography

STM. .... State Transition Matrix

UKF. .... Unscented Kalman Filter

VSD. .... Variable State Dimension filter

VSLAM. .... Visual-based Simultaneous Localization and Mapping

Abbreviations

$A(\mathbf{q})$.. Attitude matrix

a.. Sensor aspect ratio

b.. Baseline (center-to-center distance between two cameras)

$\bar{c}$ .... Mixing weights

c_x. .... Sensor optical center x-axis

c_y. .... Sensor optical center y-axis

d. .... Disparity

e. .... Image error

e_s. .... Specular exponent

f. .... Focal Length

G. .... Hessian Matrix

H. .... Measurement matrix

h. .... Angular Momentum magnitude

I. .... Identity Matrix

I. .... Moment of Inertia

I(**x**). .... Input Image

i. .... Inclination

J_2. .... Oblateness term

K. .... Gain

K_C. .... Camera Calibration matrix

K_d. .... Diffuse weighting coefficient

K_s. .... Specular weighting coefficient

L_i. .... Incident light energy

N. .... Total number of features

P. .... Camera matrix

P. .... Covariance

Pr. .... Probability Matrix

P_{VSD}. .... VSD Covariance

$\mathbf{P}_C$. .... Feature point in camera frame

Q. .... Process noise

R. .... Measurement covariance

R. .... Rotation matrix

$R_{A/B}$. .... Rotation matrix from A to B

R_e. .... Equatorial Radius

r. .... Magnitude of chief position

s. .... Skew between sensor axes

T. .... Final time

T(**x**). .... Source Image

U. .... Shanon entropy metric

u. Image coordinate x-axis

v. Image coordinate y-axis

v. Volume estimate

W. Sensor camera width

W. Warping

Vectors

$\mathbf{a}_{th}$. Thrust acceleration states

b. Gyro bias state vector

g. CRP

H. Angular Momentum

i. Unit vector x-axis

j. Unit vector y-axis

k. Unit vector z-axis

L. Torque

L. Incident light direction

m. MRP

n. Normal direction

$\mathbf{P}_I$. Feature point vector in Inertial frame

$\mathbf{p}_{i/A}$. Feature point vector from frame A

$\mathbf{p}_{i/B}$. Feature point vector from frame B

$\bar{\mathbf{p}}_{i/A}$. Centroid feature point vector from frame A

$\bar{\mathbf{p}}_{i/B}$. Centroid feature point vector from frame B

p. Consider parameter vector

$\mathbf{q}$. .... Quatenion

$\mathbf{r}_m$. .... Centroid

$\mathbf{T}$. .... Thrust states

$\mathbf{t}$. .... Translation

$\mathbf{V}$. .... Viewing direction

$\mathbf{v}$. .... Gaussian measurement random noise

$\mathbf{w}$. .... Gaussian system random noise

$\mathbf{X}$. .... ECI x-axis

$\mathbf{x}$. .... LVLH x-axis

$\hat{\mathbf{x}}$. .... State vector

$\mathbf{x}_Q$. .... Quiescent state vector

$\mathbf{x}_{VSD}$. .... VSD state vector

$\mathbf{Y}$. .... ECI y-axis

$\mathbf{y}$. .... LVLH y-axis

$\hat{\mathbf{y}}$ .... Measurement vector

$\mathbf{Z}$. .... ECI z-axis

$\mathbf{z}$. .... LVLH z-axis

$\mathbf{z}_Q$. .... Quiescent state and consider vector

$\mathbf{z}_{VSD}$. .... VSD state and consider vector

Symbols

α. .... Sensor field of view

α. .... Occupancy

$\hat{\boldsymbol{\beta}}$ .... Gyro bias estimate vector

$\delta\boldsymbol{b}$. .... Gyro bias error state vector

$\delta \boldsymbol{q}$. Quaternion error state vector

ε Threshold

Φ STM

ρ_s Surface albedo

ρ Density

Θ. Consider STM

Θ. True anomaly

μ. Gravitational Constant

$\boldsymbol{\omega}$. Angular Velocity Vector

φ. Angle between normal vector and light direction

Ψ. Mahalanobis distance

v. Volume estimate

ψ Saturation level

ACKNOWLEDGEMENTS

I would like to express my gratitude for my advisor Dr. David Spencer for teaching and guiding me through my graduate studies. His critical evaluations and guidance has been vital in my development as a researcher and engineer. I would also like to thank Dr. Puneet Singla for helping me to hone the focus of this dissertation. Furthermore, I would like to thank my remaining committee members Dr. Robert Melton and Dr. Minghui Zhu for their continued support.

I am grateful for the opportunity to have had two internships under Lt. Col. Kirk Johnson at the Air Force Institute of Technology. My work there directly contributed to the underlying basis of this dissertation. I would also like to thank Major Joshuah Hess for his guidance in my consider filter work and along with Lt. Col. Kirk Johnson, co-authored a paper together. My experiences at AFIT shaped my professional career development and led me to my current career at AFRL.

Finally, I would like to express my immense gratitude to my wife Rui. Her continued patience, love, and support during my PhD have been very encouraging and helpful. This journey would have been much more difficult without her loving support.

Chapter 1

Introduction

1.1. Motivation

Pose estimation in the space domain is a very challenging problem. These challenges result from harsh lighting conditions, computational, power and thermal constraints, and observability issues. In robotics, pose refers to the position and orientation of an object relative to the environment. The necessity for developing robust capabilities is driven by the need for autonomy to perform proximity operations under the aforementioned constraints and conditions. Systems and algorithms need to be robust enough to handle the rigors of the space domain. Operations may include servicing damaged spacecraft[1] that have become uncontrollable, refueling[2,3], building infrastructure[4], and debris removal[5–7].

In this analysis, the observer is the satellite making optical observations of the target satellite. Extensive work has been done in regards to estimation of the rotation, angular velocity, principal axes, moment of inertia, and center of mass of a spinning satellite[8–17]. Tweddle's[9] dissertation explored the estimation of motion of an uncontrolled tumbling object assuming the observer is stationary with known position and orientation. In Lichter's research[13], a measurement fusion algorithm using multiple sensors was developed to estimate the pose of a spinning satellite. The observer position relative to the target satellite was known thus reducing the problem complexity. Further studies[8,11,15] have also examined the estimation of an uncontrolled spinning satellite. However, an issue that has not seen much area of study is for a satellite exhibiting uncontrolled translational motion due to outgassing or damaged thrusters. This dissertation seeks to answer the following questions: Can an observing satellite determine its local pose in the relative frame and both detect and estimate the unknown translational motion due to an external force of an uncooperative target given that no prior information about the target's shape or the local position of the observer relative to the target is available? Can the shape of a target satellite be estimated and built from stereo vision with no a priori information and the resulting shape estimate exploited for use in improving estimation performance? In this formulation, no external forces are assumed to be acting on the target except for a low-magnitude, continuous thrust. It is also assumed that the target is not rotating though estimating its principal axes and center of mass are still desired. Two cameras are utilized to observe the target and the resulting unknown translational motion and target shape is extrapolated.

While it is reasonable that unbalanced or damaged thrusters could also impart a rotational component on the target, coupling translational and rotational motion is presents several difficulties including: image scale and rotation effects on determining pose estimation; wrongly estimating one motion for another causing issues in accurate estimation of angular velocities, moment of inertias, thrust, and maneuver detection; removing simplifying assumptions used in prior work for a rotating (non-translating) object and increasing difficulty of accurately tracking and matching image features across frames. Since a simultaneous rotation and thrusting of an uncooperative satellite with a moving observer would be an exceedingly challenging problem, this dissertation assumes the target is not rotating. Rotation of a tumbling spacecraft has been a well-studied problem making simplifying assumptions such as the observer is in a static position, multiple sensors are available, a priori information known about the target shape, or access to known state information of the observer. A comprehensive review of the available methods will be covered in the literature review.

1.2. Organization

The organization of this dissertation is as follows. Chapter 2 outlines the literature review of existing methodologies and theories pertinent to the developed algorithms. Existing concepts in dynamics, kinematics, and computer vision are reviewed for the understanding of the contributions outlined in the previous section.

Chapter 3 covers the computer vision utilized in this work in addition to the estimation algorithms. Existing approaches are outlined first so that the developed algorithms in the second half of Chapter 3 are better understood. Assumptions made and decisions for implementation of methodologies are discussed. The formulation of the synthetic imager is also covered.

Chapter 4 presents results from the developed algorithms. Two scenarios are presented with different thrust magnitudes to analyze the algorithm’s effectiveness on detecting and estimating different order of magnitude of thrust.

Chapter 5 discusses the conclusions and proposed future work. It provides scope on the presented results and overall analysis of performance. Furthermore, future avenues of work are explored.

Chapter 2

Background and Literature Review

This chapter discusses the filter approaches toward a Simultaneous Localization and Mapping (SLAM) solution. Constraints dictate what filter approaches are amenable for implementation onboard a spacecraft. Existing maneuver detection approaches are also discussed. Lastly, previous work in applying SLAM to a space-based rendezvous proximity operations problem is overviewed.

2.1. Filter-based SLAM

Autonomous rendezvous and proximity operations are of vital interest to space domain applications. Some specific applications include docking, refueling or servicing missions, proximity operations and/or landing on an asteroid, and potentially orbital debris collection. All of these missions differ in various ways but the one similarity is for the need to determine where the reference spacecraft is locally relative to the object of interest; being able to generate an accurate mapping of this position as well as a map of the object itself. This is a complex problem compounded in the situation where a priori information about the object's position and features are unknown. Within the past couple decades, researchers have strived to explore this area and SLAM was derived from terrestrial autonomous robotic applications. The basic premise of the SLAM approach is to determine the local position and orientation of a robot (pose) relative to an object or environment while simultaneously constructing a map[18,19]. As SLAM iterates, further pose information updates the local maps ultimately building a global map.

SLAM has seen many innovations over the last few decades and has expanded to many applications[18-20]. In general, the architecture of a modern SLAM problem can be split into two main components[20] (Figure **2-1**). The first part is referred to as the *front end* whose function is to take sensor data and apply them to suitable models for later pose estimation and map construction. The second, the *back end*, extrapolates the data provided by the *front end* and then feeds it back for loop closure. In a vision-based SLAM (VSLAM) application, the *front end* consists of the feature extractions from an input image of pixel locations. Data association is also done by this component and within the VSLAM example, would consist of matching the extracted features to landmarks. The *back end* models the observations and performs the pose estimation. The loop ends when observations cease. Overall, the goal of these two parts is for the autonomous object maneuvering in an unknown environment to iteratively build a map while simultaneously finding its location[18]. Durrant-Whyte and Bailey[18,19] conducted a series of surveys on the SLAM giving an excellent overview of the problem. Two general types of implementations of the SLAM solution are discussed; the filter based Extended Kalman filter (EKF) solution and the particle based Rao-Blackwellised filter, or FastSLAM. An EKF-SLAM formulation has been well studied and exhibits the advantages and disadvantages associated to EKF based solutions for navigation and tracking.

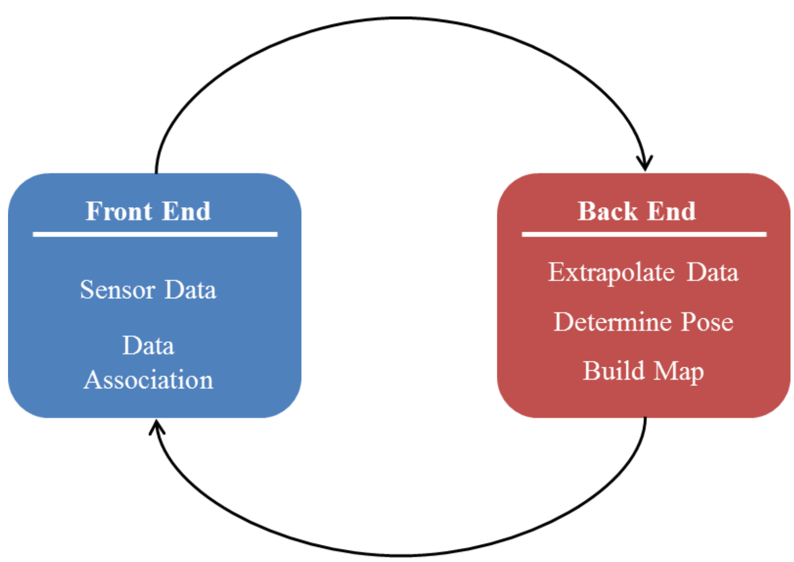

Figure **2-1**: General SLAM architecture

A popular approach for SLAM is using the Maximum a Posteriori (MAP) formulation often using factor graphs to reason the connection between variables. Unlike Kalman filtering, MAP estimation is not reliant on any distinction between motion and observation models as both are treated as factors[20]. Factor graphs represent interdependence between the various variables and factors through encoded nodes on the overall graph. Unlike filter-based approaches, factor graphs enable loop closure and do not require linearization of measurement updates[9] allowing for general improved performance. This methodology is very useful and generally more robust then an EKF-SLAM at some extra computational expense and has been applied to many different applications[9,10,21–24].

The shortcomings of the EKF-SLAM have been well documented in the literature[25–28]. Inconsistency in the estimation arises between the linearization made by the EKF, the inherent discrepancy from the true nonlinear system, and the observations. However, much work has been done to overcome these limitations and to improve their robustness. One such approach is the Iterated Extended Kalman Filter (IEKF) which has been used to produce good results[15,26,29]. The premise of the IEKF is that the estimated states are iteratively updated once the observations have been applied and re-linearized. This new state estimate has higher accuracy over the standard EKF and produces better Jacobians (first-order partial derivatives of system dynamics with respect to the state variables) in the next time step.

Other methodologies for improving upon or replacing the EKF have been proposed as well. The smooth variable structure filter[32,33] uses variable structure theory and sliding mode concepts. It has been shown to be robust and consistent when applied to an unknown system. Another approach involves eliminating the computation of covariance matrices altogether by using a small set of ensemble members that define the state[34]. The Ensemble Kalman Filter is similar to a particle filter in that the ensemble members are a set of points (ensembles) that are propagated instead of the state and covariance. The mean of the ensemble represents the state estimates and the sample covariance represents the covariance.

Hybridization of techniques has also been explored. Augenstein[8] proposed a hybridized methodology for SLAM in which rotational and translational pose estimation are conducted simultaneously via Bayesian filtering and optimization respectively. Furthermore, Leishman[22,23] developed an approach where a multiplicative extended Kalman filter (MEKF) is intertwined with a graph-based SLAM. Estimates from the MEKF are kept relative to the nodes in the factor graphs instead of the global reference frame as done in a classical MEKF. Utilizing this hybridized approach minimizes the needs that using the global states would entail. The advantage of this method allows vision-based observations to be used directly and eliminates the need for feedback to the filter from loop closure which is a computationally expensive process.

Improving computational efficiency is highly desired in real-time applications. However, this frequently comes at the cost of consistency and accuracy. One key element within some of the aforementioned approaches is the sparsification of matrices[19,29–31]. Sparsification refers to making elements much smaller than *1.0* set to *0*. An EKF-SLAM represents a state estimate and covariance matrix. The state estimates comprise the desired parameters to be estimated and their associated uncertainty referred to as the covariance. Alternatively, the information matrix, which is the inverse of the covariance matrix, is used in place of the covariance. Through using this formulation, the off-diagonal terms of the normalized information matrix are close to zero. This can be exploited in a sparsification procedure that sets all these approximately zero components to zero. The advantage of this newly sparse matrix is the savings in computation with a little loss in accuracy[19].

In this dissertation, a filter-based solution was chosen due to computational limitations onboard spacecraft. While other approaches discussed have greater robustness, they are computationally expensive and are better suited for offline estimation or terrestrial-based applications.

2.2. Maneuver Detection

Objects classified within the space domain include both actively controlled non-cooperative and cooperative objects as well as orbital debris[35]. In the case of actively controlled non-cooperative objects, detecting unknown maneuvers is a part of the overall tracking and statistical orbital determination problem. Many approaches exist for both detecting and estimating an unknown maneuver. Specifically, many variations and extensions of the Kalman Filter have been successfully implemented and applied to maneuver detection[35–41].

Detecting and estimating maneuvers can be a challenging problem for a variety of reasons. Measurement and process noise can potentially mask the detection of a maneuver depending on their levels. Inconsistencies in the dynamic model and the linearity assumptions from the Kalman Filter can also adversely affect the ability to detect and estimate maneuvers. Duration and magnitude of the thrusting maneuver itself is correlated to the success of the filter. Lower thrust magnitudes can be disguised by noises inherent in the system and short durations can make it difficult for the solution to converge.

Filter-based solutions have approached this problem in several ways. One method applies a smoothing filter to utilize current measurement updates to check on the consistency of past states; a maneuver is indicated if the consistency test fails[35,38,41]. An example of a smoother approach is the Rauch-Tung-Striebel smoother (RTS) which is an efficient way of smoothing as it does not need to directly compute the backward estimate or covariance[42,43]. This formulation starts by executing a normal Kalman filter forwards in time up to a final time T. Once this time is reached, the RTS is initialized by setting the smoothed state and covariance at time T as the current state and covariance estimates respectively. It starts at time equal to T and iterating backwards toward the start of the forward Kalman filter. The filter test operates by defining the difference between the filter and smoother covariance. Thus, the parameter performing the consistency test is related to the aforementioned covariance If any element of this parameter is above a predefined threshold, the consistency test would fail.

A failure in the smoothing consistency indicates a change in the dynamics (from natural dynamics with no external forces to dynamics consisting of acceleration terms due to thrusting), and thus a maneuver is assumed to have occurred. This approach is fairly robust, but still has a couple of limitations. Since smoothers work by running backwards on previous batches of data, special care must be taken when implementing in real time. The defined window in which the smoother processes a batch also affects its effectiveness on maneuver detection. A maneuver occurring outside this window will not be detected until the next window. The length of these windows is dependent on the frequency of observations.

Another general approach to maneuver detection is the Variable State Dimension filter (VSD)[35,36]. This method works by estimating the nominal states until the residual grows (after a maneuver occurs) to a point where a threshold is surpassed. This threshold is chosen based on the chi-squared test. Passing this pre-defined threshold initiates a model switch, and the filter concatenates the thrust states onto the nominal state vector and estimates them directly. Advantages of the VSD include the ability to run in real-time due to its low computational cost. However, it is sensitive to the threshold value; depending on the process and measurement noises present, either a maneuver may go undetected or conversely a false maneuver will be detected. Results obtained are also dependent upon the lag between a maneuver occurring and the detection of the maneuver.

Multiple model filters have been used in many estimation applications, and several have applied this methodology to maneuver detection[35,39,40]. The main advantage of applying multiple model filters is that the state estimation is not dependent on any one solution; rather, the presence of multiple models allows for multiple solutions to be weighed and combined, used as feedback to improve the others, or chosen amongst the other solutions as the most viable option. However, the drawback of this general approach is the expensive computational cost. Weighting the results from multiple models and combining the state estimates provides robust correction to the update which may make the presence of consider parameters superfluous.

Another approach developed by Lubey and Scheeres[99,100] utilizes an Optimal Control Based Estimator (OCBE) with additional algorithms for maneuver detection. The OCBE is essentially a generalized Kalman filter with dynamic noise propagation (equivalent to continous process noise) and smoothing properties. This algorithm fuses dynamics and state estimation through optimal control. Output from this algorithm includes the control policy which identifies mismodeled natural dynamics which is used in an additional algorithm to detect maneuvers based on the level of dynamic uncertainty.

The VSD approach combined with multiple models forms the basis of the algorithm developed in this dissertation. VSD filters provide a low computational cost method of thrust detection and estimation. Multiple models are only used when the maneuver is detected and switches back to a single model approach when a maneuver is no longer active. This reduces computational costs outside of the maneuver estimation and provides the ability for the filter to cope with higher uncertainty during a maneuver. Utilizing a limited number of models keeps computations low during a maneuver while also providing further robustness.

2.3. Space Domain SLAM

A SLAM-based approach has provided solutions to several space-domain problems. Tweddle's dissertation[9] and resulting paper[10] develops a factor graph approach in which a six degree-of-freedom rigid-body dynamics system estimates the position, attitude, linear and angular velocities, center of mass, principal axes and moment of inertia parameters. An incremental smoothing and mapping (iSAM) is modified find these estimates using a feature-based stereo-vision observation model. Properties pertaining to the orientation of a target object are found from a static observer position with known camera position and orientation. Contributions made to estimating a target object's properties include a parametrization approach for estimating center of mass and the principal axes by use of a geometric frame in which the stereo-vision feature points are estimated. Furthermore, the principal moment of inertias is parametrized by taking the natural logarithm of the ratio of elements in the principal moment of inertia matrix. Estimating the moment of inertias is not observable and can only be recovered up to a scaling factor. The approach developed by Tweddle[9,10] is limited to an offline process as the algorithm estimates the set of stereo-vision feature points for the purposes of loop-closure and improving state estimates. Using the mapping estimated by this framework, the principal axes and moment of inertias are recovered up to a scaling factor by using MeshLab. Feature points creating the target mapping are included in the state vector increasing the computational cost of the algorithm. However, including the feature points comprising the map allows for loop closure.

In a paper by Lavagna et al,[12] relative position, velocity, angular velocity, attitude and inertial properties of an uncooperative spacecraft are estimated by an Iterated Extended Kalman Filter (IEKF) using only stereo-vision measurements. The developed observation model utilizes pixel coordinates along with exploiting the optical flow (velocity of the moving pixels between frames). A pseudo measurement constraint was included in the measurement model to force the convergence of the inertia matrix.

The inertia matrix itself is parametrized using an approach developed by Tweddle[9,10] in which the first and third elements of the principal inertias are normalized by the second element and subjected to the natural logarithm. This procedure reduces the number of variables to the degrees of freedom. By using the natural logarithm, the two normalized parameters fit within the same domain as the natural logarithm itself. The entire inertia matrix is reconstructed in an offline process by creating a triangular mesh from a sequence of point cloud data to produce the model geometry of the object. By assuming density is constant, an estimate of the inertia parameters is found. Mass properties of the object are found in a similar manner assuming a uniform density distribution.

Lichter's work[13,14] solves a similar problem to Tweddle in pose information of a tumbling spacecraft is extrapolated from optical observations. In Lichter's dissertation, observations from multiple sensors (stereo cameras, laser range finder) are fused together to form the basis of a coarse target attitude and center of mass estimation. From these fused measurements, a voxel map is created to provide coarse approximations for the geometric principal axes and center of mass. The voxel map is simultaneously built with no a priori information about the target shape. The principal axes and center of mass estimates are used as the measurement model in an EKF framework that separates the rotational and translational (target translates under natural motion only) estimates of the target in the inertial frame into two separate filters. A quaternion is used to represent the attitude estimates as well as the inertia parameters even though the attitude has three degrees-of-freedom and the inertia parameters only two. The set of feature points comprising the mapping are not estimated which lowers the computational costs but does not provide loop closure and can lead to smearing of the map.

Research conducted by Augenstein[8] again focused on a tumbling spacecraft instead using a static monocular camera. Assuming no priori information is available, a hybrid algorithm combining Bayesian estimation methods and nonlinear optimization techniques for pose estimation. A modified Rao-Blackwellized particle filter is used to estimate the rotational dynamics of the target object assuming the rotation is modeled as a Gaussian process. Rigid body dynamics were not incorporated thus not estimating the moment of inertia.

Sharma and D'Amico[11] presented a paper for a monocular based pose estimation. Pose of a passive target is estimated using a known three-dimensional wireframe model with no priori information on range measurements or state estimates. Due to the range ambiguity of monocular vision, a pose initialization approach was developed which uses the wireframe model with no other assumptions on translational or rotational information. A weak gradient technique is implemented that pairs the wireframe model with the two-dimensional monocular images to create multiple correspondence hypotheses. Multiple pose solutions are determined from this process and are refined iteratively by using the Newton Raphson Method. Their navigation filter utilizes a MEKF with a measurement model that detects line segments in the images. Using line segments makes the navigation filter more robust to illumination effects and helps to distinguish the boundaries of the spacecraft from the background.

Feng et al[16] developed a methodology to estimate relative pose information and inertial properties of a target. The approach consists of establishing a body-fixed frame to the target using the surface features and measures the relative attitude; a standard Kalman filter is implemented to estimate the translational states and center of mass; and an EKF and UKF to estimate the rotational states and moment of inertia ratios respectively. The moment of inertia is parametrized by dividing all elements by I_{xx}.

Estimation of a tumbling object in an autonomous proximity operation is a rich area of study. However, an area of limited research pertains to unknown, uncontrolled translational maneuvers within an autonomous proximity operation using visual observations. Furthermore, extrapolating mass information from the shape model is limited to an offline process in the literature (Tweddle[9,10] and Lavagna et al[12,15]). Model shape estimates are input into a robust mesh-based software to extrapolate more refined principal axes and mass estimates. Leveraging an online mass estimator is something that is not present in the literature and can provide useful utility for a maneuver estimation algorithm.

In this dissertation the relative position of the inspector to the target is estimated in addition to the rotation between camera frames. The inspector needs to determine its localization in relation to the target such that it can develop an accurate shape estimate of the target. This shape estimate is exploited such that it can prune erroneous measurements from entering the filter as well as providing information directly to the estimation filter to aid in maneuver estimation. Parameters are extrapolated from the target shape map that are used in the estimation filter and guidance control algorithm. Pose of the inspector is estimated simultaneously with the target shape. The target executes an unknown maneuver with no information regarding thrusting magnitude and thrusting duration. Detailed contributions towards this endeavor are in Section 2.4.

2.4. Contributions

This dissertation will make the following contributions toward the implementation of autonomous navigation systems onboard a spacecraft. These contributions are based on gaps in the current state-of-the-art as described in Sections 2.1-2.3. Areas of limited study include a low computational onboard process of visual maneuver detection and development of an online process to exploit shape estimation to extrapolate target characteristics that can improve the maneuver detection and pose estimation.

1) A computational optimal method using consider parameters and a variable state dimension (VSD) filter is developed to detect and estimate thrust states of an uncooperative spacecraft with no a priori information of the thrusting start/end time. With little accuracy sacrificed to existing maneuver detection methods, this approach will allow onboard implementation.
2) Using stereo-vision (two or more cameras) observations, an interactive multiple model octree-based approach provides an onboard coarse estimation of volume and thus mass assuming an evenly distributed density to be used in conjunction with the maneuver detection algorithm.
3) Demonstrate how simultaneous localization (position and orientation of the inspector in the relative frame) and octree mapping (target shape estimation) provides information on target center of mass and principal axes with no priori information by exploiting mapping to improve measurements into the system once the confidence in the mapping reaches a pre-defined threshold.

A synthetic image generation framework is developed as a test-bed to create a sequence of images of a model spacecraft to provide measurements to the aforementioned estimation filters. The localization and target mapping algorithms are separated and together form the basis of this dissertation.

Chapter 3

Mathematical Formulation, Computer Vision and Filter Approach

This chapter details the necessary concepts to formulate the developed algorithm in the later parts of the chapter. These include rigid body kinematics, stereovision, quaternions, and octrees. The filter approaches and developed algorithms are explained in detail. Creation of the observation system is also discussed.

3.1. Rigid Body Dynamics

The rigid body equations with respect to an inertial frame are derived from Newton's second law. Angular momentum can be written as

`

$$\mathbf{H} = \mathrm{I}\boldsymbol{\omega} \tag{3.1}$$

where I is the moment of inertia and $\boldsymbol{\omega}$ is the angular velocity. With respect to the inertial frame, the rate of change of angular momentum is equal to the torque $\mathbf{L}$. Using the transport theorem leads to

$$\mathrm{I}\dot{\boldsymbol{\omega}} = \dot{\mathbf{H}} = \frac{\mathrm{d}}{\mathrm{dt}}(\mathbf{H}) + \boldsymbol{\omega} \times \mathbf{H} = \mathbf{L} \tag{3.2}$$

Euler's equation of motion on a rigid body can be derived from Equations 3.1 and 3.2 yielding

$$\begin{gathered}\mathrm{I}\dot{\boldsymbol{\omega}} = -[\boldsymbol{\omega}\mathrm{x}]\mathrm{I}\boldsymbol{\omega} + \mathbf{L} \\ \boldsymbol{\omega}\mathrm{x} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}\end{gathered} \tag{3.3}$$

where the $[\boldsymbol{\omega}\mathrm{x}]$ term is the skew of the angular velocity. These dynamics are utilized by the inspector spacecraft in a non-linear Lyapunov control law detailed in Section 3.18.4.

3.2. Stereovision

A common model used for a camera is the pinhole model. This model is depicted in Figure **3-1**. Consider a point $\mathbf{P}_I$ in the inertial frame which can be projected onto the image plane with coordinates *u* and *v*; these coordinates reside in the two dimensional image plane. The camera itself has a focal length *f*, which can be expressed in terms of millimeters or pixel length. In the image plane, the optical center is denoted by c_x and c_y. This is the standard perspective geometry for the pinhole camera model.

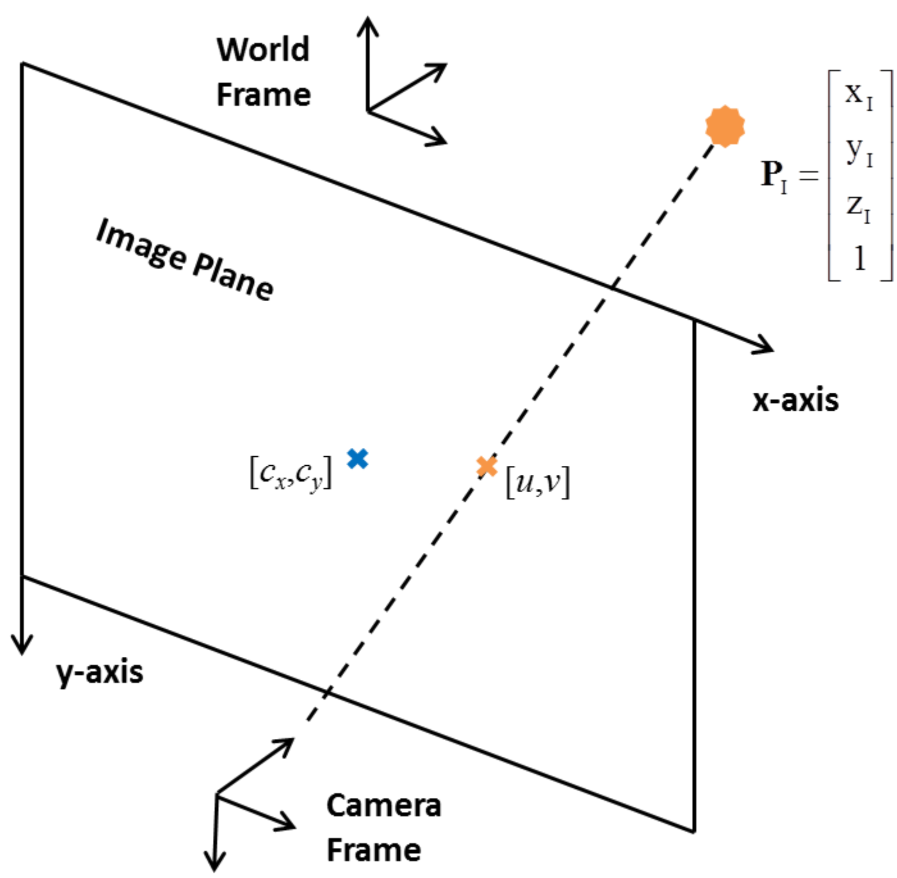

Figure **3-1**: Pinhole Camera Model

Homogeneous coordinates can be especially helpful in defining useful properties in the image processing model[44]. They are a system of coordinates frequently used in projective geometry as they can express points at infinity using finite coordinates. An extra dimension is required of the projective space considered. A 2D homogeneous point can be defined by

$$\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{w})^T = \tilde{w}(x, y, 1)^T = \tilde{w}\bar{\mathbf{x}} \tag{3.4}$$

where $\tilde{x}$, $\tilde{y}$, and $\tilde{w}$ are the homogenous coordinates and $\bar{\mathbf{x}}$ is the augmented vector made up of the imhomogeneous coordinates and an additional coordinate containing the scalar value of *1*. The imhomogeneous coordinates are related to the homogeneous coordinates by dividing through by $\tilde{w}$. In a similar manner, homogeneous coordinates can be extrapolated to 3D coordinates. This is written as

$$\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w})^T = \tilde{w}(x, y, z, 1)^T = \tilde{w}\bar{\mathbf{x}} \tag{3.5}$$

Camera properties can be split into two categories: camera intrinsics and extrinsics[44]. The intrinsics describe the camera's optical properties and the extrinsics refer to the orientation of the camera. Rotation and translation between the inertial and camera frame can be expressed by a rotation matrix R and a translation vector **t**. Augmenting the translation vector to the rotation matrix yields a *3*x*4* matrix.

$$\left[\mathrm{R} \mid \mathbf{t}\right] \tag{3.6}$$

These characteristics represent the extrinsics of the camera's properties. Together with the calibration matrix, a camera matrix can be defined as

$$\mathrm{P} = K\left[\mathrm{R}|\mathbf{t}\right] \tag{3.7}$$

where P is the camera matrix of size *3*x*4*. A *4*x*4* invertible matrix can be used as well which is written as

$$\mathrm{P} = \begin{bmatrix} K & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{3.8}$$

The calibration matrix K can be defined as

$$K = \begin{bmatrix} f & s & c_x \\ 0 & af & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

where f is the focal length, c_x and c_y are the coordinates of the optical center, a is the aspect ratio, and s is the skew between the sensor axes due to improper alignment relative to the optical axis. A similar form of the calibration matrix exists by setting $a = 1$ and $s = 0$. Commonly, the optical center of the image is set to half the sensor width and height. Field of view can be defined as

$$\alpha = 2\tan^{-1}\left(\frac{W}{2f}\right) \tag{3.10}$$

where α is the field of view, and W is the camera width.

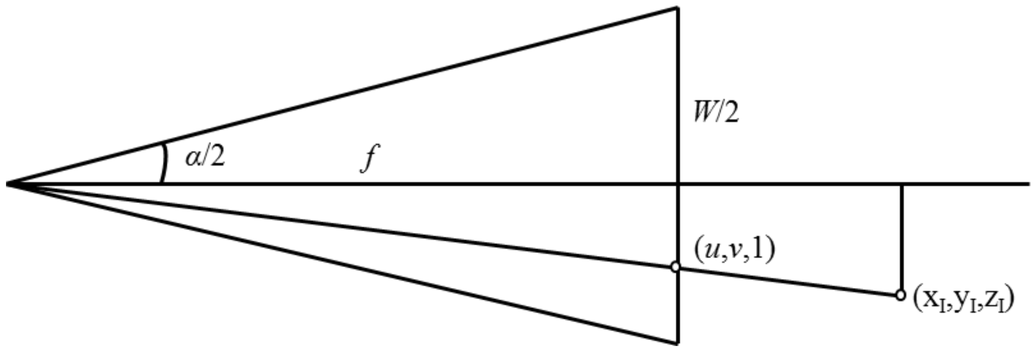

Figure **3-2**: Camera projection of inertial coordinates on to image frame

A calibration matrix *K* can be defined describing the camera intrinsics. The *4*x*4* camera matrix can transform the inertial coordinates directly to the image coordinates. Inertial and image coordinates can be denoted as

$$\mathbf{P}_I = (x_I, y_I, z_I, 1) \tag{3.11}$$

$$\mathbf{P}_{\mathrm{Im}} = (u, v, 1, d) \tag{3.12}$$

where $\mathbf{P}_{\mathrm{I}}$ and $\mathbf{P}_{\mathrm{Im}}$ are the inertial and image coordinates respectively, and *d* is the disparity. Mathematically, the relationship between the inertial and image coordinates is given by

$$\mathbf{P}_{\mathrm{Im}} = P\mathbf{P}_I \tag{3.13}$$

Using the *3*x*4* camera matrix, the formulation changes slightly as follows

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & s & c_x \\ 0 & af & c_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}_C \tag{3.14}$$

$$\mathbf{P}_C = [\mathrm{R} \mid \mathbf{t}]\mathbf{P}_I = [\mathrm{R} \mid \mathbf{t}]\begin{bmatrix} x_I \\ y_I \\ z_I \\ 1 \end{bmatrix} = \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} \tag{3.15}$$

where $\mathbf{P}_{\mathrm{C}}$ is the coordinates in the camera frame. Here, the image vector contains one less component than Equation 3.12. The projection of the inertial coordinates on the image plane is visualized in Figure **3-2**.

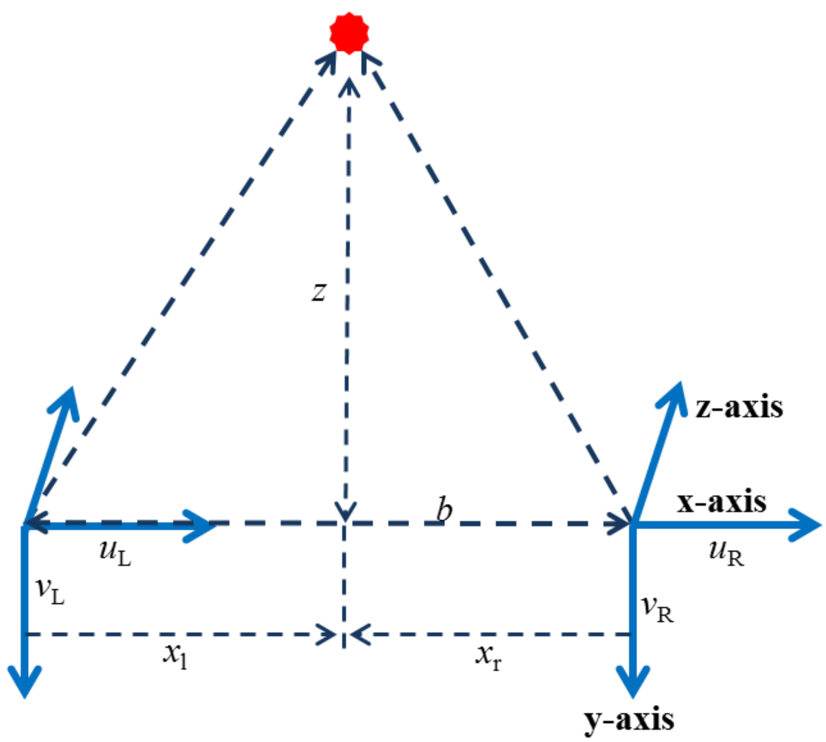

Figure **3-3**: Stereovision Geometry

Utilizing two or more cameras is referred to as stereo vision. It possesses inherent advantages in reducing the range ambiguity issue over monocular cameras due to the ability to triangulate a three dimensional position from multiple views. Geometry of the two cameras can be seen in Figure **3-3**.

Both the left and right cameras have image coordinates [u_l v_l] and [u_r v_r] respectively. The baseline b is defined as

$$b = x_l - x_r \tag{3.16}$$

where x_l and x_r are the distances along the camera's unit **x** vector. A metric for the distance between two pixel pairs in the image frame is defined as the disparity. The pixel locations in the image plane is related to the disparity, d, by

$$d = \sqrt{(u_l - u_r)^2 + (v_l - v_r)^2} \tag{3.17}$$

Normally, calibration of the cameras leads to the fact that $v_l = v_r$. This simplification reduces the definition of disparity to

$$d = u_l - u_r \tag{3.18}$$

Through the geometry established in Figure **3-3**, the following relationships can be derived as

$$\frac{x_l}{z} = \frac{u_l}{f} \tag{3.19}$$

$$\frac{x_r}{z} = \frac{u_r}{f} \tag{3.20}$$

where f is the focal length and z is the boresight of the camera. The depth to these camera frame coordinates is given by

$$z = \frac{fb}{d} \tag{3.21}$$

Combining Equations (3.19)-(3.21) and rearranging terms yields the camera frame coordinates

$$x_l = \frac{u_l}{f} z = \frac{u_l b}{d} \tag{3.22}$$

$$x_r = \frac{u_r}{f} z = \frac{u_r b}{d} \tag{3.23}$$

$$y_l = \frac{v_l}{f} z = \frac{v_l b}{d} \tag{3.24}$$

$$y_r = \frac{v_r}{f} z = \frac{v_r b}{d} \tag{3.25}$$

Within the scope of SLAM problem, these coordinates represent the landmark feature locations and will encompass the measurements sent to the pose estimation.

3.3. Feature Detection and Matching

As described in the previous section, features from the environment are projected into a *2*-dimensional image plane. Through the stereopsis relationships in Equations (3.22)-(3.24), these image features are triangulated into *3*-dimensional space in the camera frame. These features in the camera frame form the basis of the measurement model as described in Section 3.6. Given an image, these *2*-dimensional image points must first be detected and extracted. The Kanade Lucas Tomosi (KLT) is a feature tracker with the ability to track feature matches through continuous frames[45]. This algorithm as seen many updates and optimizations over the years since its inception in 1981 by Lucas and Kanade[46]. Fundamentally, the goal is to minimize error between a source image and an input image and to map that image back onto the source image. Let $T(\mathbf{x})$ denote the source image and $I(\mathbf{x})$ the input image where x is a 2D vector of the image coordinates u and v. Lastly, define a mapping referred to as warps as $\mathbf{W(x;p)}$ where

$$\mathbf{p} = (p_1, p_2, ... p_n)^T \tag{3.26}$$

in which $\mathbf{p}$ is an n-dimensional vector containing parameters. Warping is the map between a pixel coordinate in $\mathbf{x}$ of the source image frame and maps it to the corresponding point in the input image frame. Minimizing the error (e) between the warped input image $I(\mathbf{x})$ and the source image is accomplished by

$$\mathrm{e} = \sum_{\mathrm{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - T(\mathbf{x}) \right]^2 \tag{3.27}$$

Minimization is conducted with respect to the parameters **p** and it is assumed that the current estimate of **p** is known and proceeds to iteratively solve for incremental changes in the parameters. Equation (3.28) can thus be rewritten as

$$e + \Delta e = \sum_{\mathrm{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p} + \Delta\mathbf{p})) - T(\mathbf{x}) \right]^2 \tag{3.28}$$

where $\Delta\mathbf{p}$ is the incremental change in the parameters and Δe is the incremental change in the error. The parameters are updated with this incremental change iteratively until they converge. Normally, this is done until the norm of $\Delta\mathbf{p}$ is below some predefined threshold. Linearizing Equation 3.28 through a Taylor expansion yields

$$e + \Delta e = \sum_{x} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \nabla I \frac{\partial W}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right]^2 \tag{3.29}$$

where the gradient of the input image I is calculated in the input image frame and then warped to the source image frame through the mapping. Additional formulations are needed to form the basis of the algorithm. Taking the partial derivative of Equation (3.29) with respect to $\Delta\mathbf{p}$ yields

$$\frac{\partial(e + \Delta e)}{\partial \Delta\mathbf{p}} = 2\sum_{x} \left[\nabla I \frac{\partial W}{\partial \mathbf{p}} \right]^T \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) + \nabla I \frac{\partial W}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x}) \right] \tag{3.30}$$

Taking Equation (3.30) setting it equal to zero, and solving for $\Delta\mathbf{p}$ gives the solution for the minimum of Equation (3.29) which is written as

$$\Delta\mathbf{p} = G^{-1}\sum_{x}\left[\nabla I\frac{\partial W}{\partial \mathbf{p}}\right]^{T}\left[T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))\right] \tag{3.31}$$

where G is a Hessian matrix defined as

$$G = \sum_{x}\left[\nabla I\frac{\partial W}{\partial \mathbf{p}}\right]^{T}\left[\nabla I\frac{\partial W}{\partial \mathbf{p}}\right] \tag{3.32}$$

Using the preceding equations, the algorithm for the KLT is displayed in Figure **3-4**.

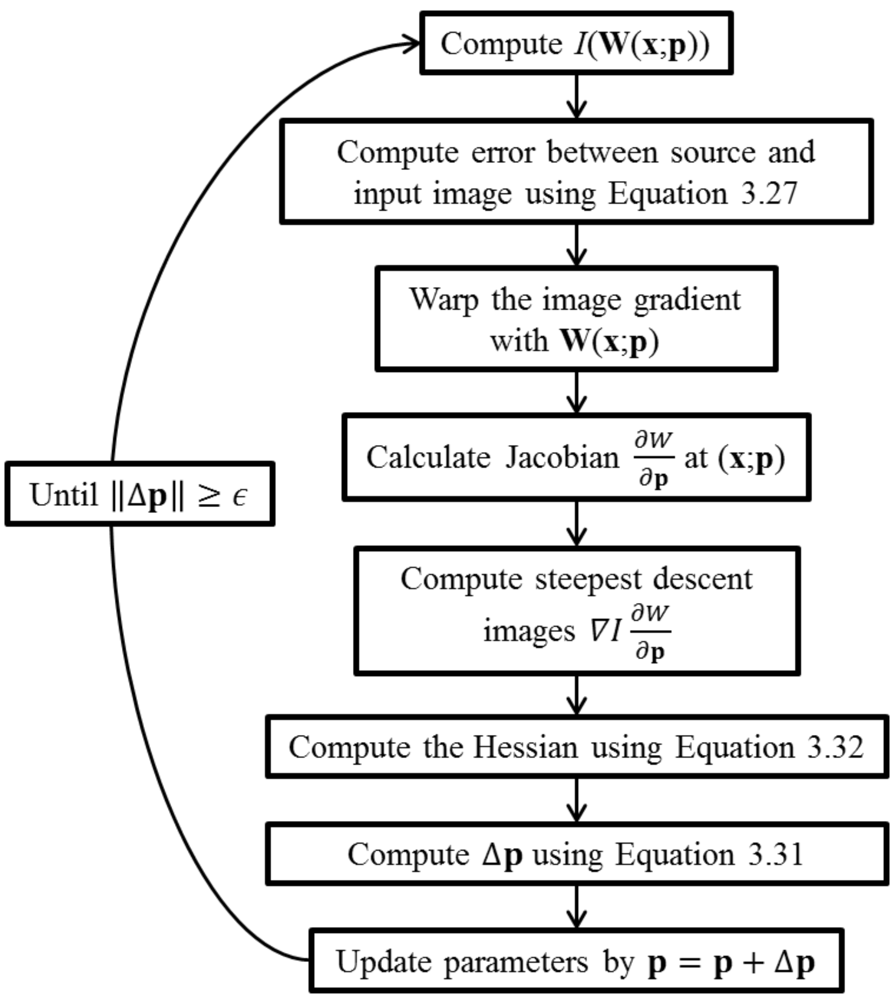

Figure **3-4**: The Lucas Kanade Algorithm[47]

As previously mentioned, there have been many improvements upon this original formulation. A more detailed derivation of the original algorithm as well as derivations of more robust versions are presented in the Baker and Matthews paper[47]. The developed algorithms show improved results over the original formulation and perform similarly amongst themselves. Choice of the algorithm depends whether more noise is present in the source image or the input image and the importance of the algorithm optimization.

3.4. Relative Dynamics

There are many dynamical models that have been used within the realm of the local vertical local horizontal (LVLH) frame. In an excellent survey by Sullivan et al[48], several relative models are evaluated based on metrics including but not limited to their performance and accuracy. The model developed by Xu and Wang[49–51] was incorporated in the estimation filter as the dynamics are nonlinear and contain the J_2 oblateness term. It is common in the LVLH frame to define both a chief and deputy satellite. The chief is not moving within the LVLH frame (though the frame itself is moving in inertial coordinates) and the center of the frame is assumed to be the center of mass of the chief. The deputy refers to the satellite that is trying to perform some rendezvous proximity operations relative to the chief. Definition of the axes follows Vallado's[52] definition of the LVLH frame. The **x** axis is aligned with the radial vector of the Earth with the positive direction pointing towards the center of the Earth. The **y** axis is in the direction of the angular momentum and the **z** axis completes the triad.

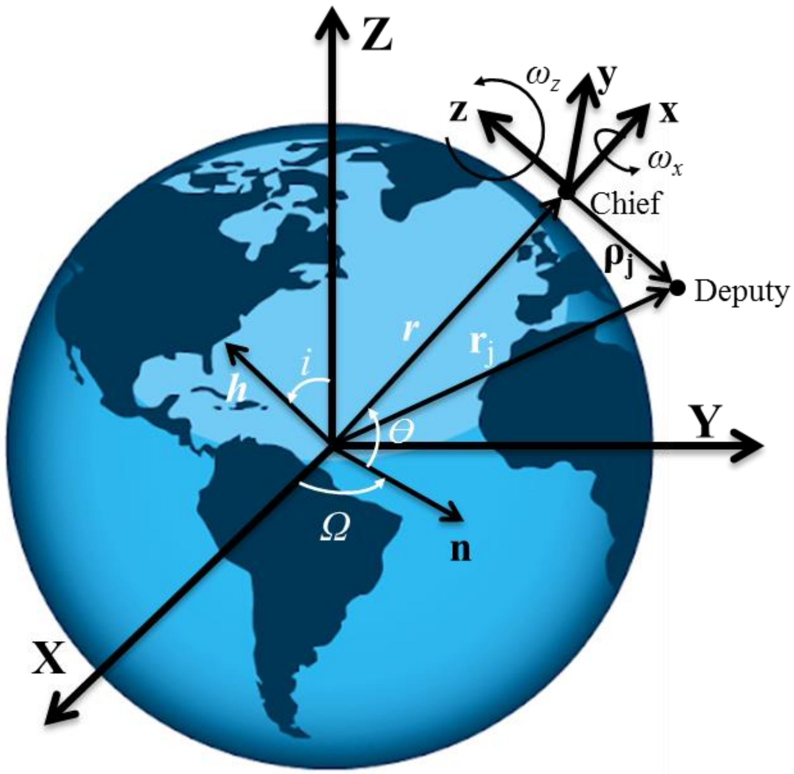

Figure **3-5**: ECI and LVLH frames

The origin of the LVLH frame is centered on the chief satellite as shown in Figure **3-5** where the angular velocity of the frame is defined as:

$$\varpi = \omega_x \mathbf{x} + \omega_z \mathbf{z} \tag{3.33}$$

where the individual components ω_x and ω_z are referred to as the steering rate of the orbital plane and the orbital rate respectively. The term ω_y is equal to zero. They can both be computed as:

$$\omega_x = -\frac{k_{J2} s_{2i} s_\theta}{h r^3} \tag{3.34}$$

$$\omega_z = h / r^2 \tag{3.35}$$

where h and r are the magnitudes of the chief's angular momentum and geocentric radius in the ECI (Earth centered inertial frame with origins centered on the Earth) frame. The s with subscripts i and Θ denote the sine of the angles of inclination and true anomaly respectively. Furthermore, the subscript *2* means twice the angle. The constant k_{J2} appears in many of the following equations to group repeatedly used constants and is defined as:

$$k_{J2} = 3J_2 \mu R_e^2 / 2 \tag{3.36}$$

The constant R_e is the equatorial radius of the Earth and μ is the gravitational constant. Utilizing those formulations, the differential equations in the ECI frame describing the motion of the chief satellite's orbit can be described by:

$$\dot{r} = v_x \tag{3.37}$$

$$\dot{v}_x = -\frac{\mu}{r^2} + \frac{h^2}{r^3} - \frac{k_{J2}}{r^4}\left(1 - 3s_i^2 s_\theta^2\right) \tag{3.38}$$

$$\dot{h} = -\frac{k_{J2} s_i^2 s_{2\theta}}{r^3} \tag{3.39}$$

$$\dot{\theta} = \frac{h}{r^2} + \frac{2k_{J2} c_i^2 s_\theta^2}{hr^3} \tag{3.40}$$

$$\dot{i} = -\frac{k_{J2} s_{2i} s_{2\theta}}{2hr^3} \tag{3.41}$$

$$\dot{\Omega} = -\frac{2k_{J2} c_i s_\theta^2}{hr^3} \tag{3.42}$$

Xu and Wang defined the five variables (v_x, h, r, Θ, i) as the compact reference satellite variables (CRSV). These variables define the motion of the chief and will be directly used in the derivation of the rotation of the LVLH frame. Taking the time derivative of Equations (3.33) and (3.34) and substituting the variables of the CRSV into the equations yields the following:

$$\alpha_x = \dot{\omega}_x = -\frac{k_{J2} s_{2i} c_\theta}{r^5} + \frac{3 v_x k_{J2} s_{2i} s_\theta}{r^4 h} - \frac{8 k_{J2}^2 s_i^3 c_i s_\theta^2 c_\theta}{r^6 h^2} \tag{3.43}$$

$$\alpha_z = \dot{\omega}_z = -\frac{2 h v_x}{r^3} - \frac{k_{J2} s_i^2 s_{2\theta}}{r^5} \tag{3.44}$$

The variables of α_x and α_y are the steering acceleration and the orbital acceleration respectively. Before the differential relative dynamic equations can be defined, a few more variables need to be computed. In the ECI frame, the position vector of the deputy satellite (satellite being tracked) is defined by the equatorial radius of the chief satellite plus the relative distance between both spacecraft ($\boldsymbol{\rho}_\mathbf{j}$):

$$\mathbf{r}_j = \mathbf{r} + \boldsymbol{\rho}_\mathbf{j} = r\hat{\mathbf{x}} + \left(x_j \hat{\mathbf{x}} + y_j \hat{\mathbf{y}} + z_j \hat{\mathbf{z}} \right) \tag{3.45}$$

Taking the magnitude of the deputy position vector gives the following equation:

$$r_j = \left| \mathbf{r}_j \right| = \sqrt{\left(r + x_j \right)^2 + y_j^2 + z_j^2} \tag{3.46}$$

The projection of the deputy position vector on the **Z** axis of the ECI frame is derived to be:

$$r_{jZ} = \mathbf{r}_j \cdot \hat{\mathbf{Z}} = \left(r + x_j \right) s_i s_\varphi + y_j s_i c_\varphi + z_j c_i \tag{3.47}$$

Finally, all the necessary variables have been defined to now write the governing nonlinear differential equations of motion as:

$$\ddot{x}_j = 2\dot{y}_j\omega_z - x_j\left(\eta_j^2 - \omega_z^2\right) + y_j\alpha_z - z_j\omega_x\omega_z - \left(\zeta_j - \zeta\right)s_i s_\varphi - r\left(\eta_j^2 - \eta^2\right) + F_{jx} \tag{3.48}$$

$$\ddot{y}_j = -2\dot{x}_j\omega_z + 2\dot{z}_j\omega_x - x_j\alpha_z - y_j\left(\eta_j^2 - \omega_z^2 - \omega_x^2\right) + z_j\alpha_x - \left(\zeta_j - \zeta\right)s_i c_\varphi + F_{jy} \tag{3.49}$$

$$\ddot{z}_j = -2\dot{y}_j\omega_x - x_j\omega_x\omega_z - y_j\alpha_x - z_j\left(\eta_j^2 - \omega_x^2\right) - \left(\zeta_j - \zeta\right)c_i + F_{jz} \tag{3.50}$$

The various control forces are denoted by F and the accelerations of ζ and the velocities of η are as follows

$$\zeta = \frac{2k_{J2}s_i s_\theta}{r^4} \tag{3.51}$$

$$\zeta_j = \frac{2k_{J2}r_{jZ}}{r_j^5} \tag{3.52}$$

$$\eta^2 = \frac{\mu}{r^3} + \frac{k_{J2}}{r^5} - \frac{5k_{J2}s_i^2 s_\theta^2}{r^5} \tag{3.53}$$

$$\eta_j^2 = \frac{\mu}{r_j^3} + \frac{k_{J2}}{r_j^5} - \frac{5K_{J2}r_{jZ}^2}{r_j^7} \tag{3.54}$$

A more rigorous derivation of the preceding equations can be found in Xu and Wang.[49]

In this dissertation, the relative dynamics are used to express the relative position and velocity between the chaser (inspector) and chief (target). Stereovision defined in Section 3.2 is used to extrapolate *3*-dimensional features in the camera frame and applying a transformation between the camera and world frame (LVLH) leads to the relative states expressed in the equations of motion (Equations 3.48-3.50).

3.5. Rotational Kinematics

Quaternions are part of the four-tuple of real numbers consisting of four parameters[53] that are useful in representing rotations due to their non-singularity, however, they have one constraint that must be maintained. Three of these parameters are vectors and one part is a scalar. The quaternion can be written in general form as

$$\bar{\mathbf{q}} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{1:3} \\ q_4 \end{bmatrix} \tag{3.55}$$

where q_1, q_2, and q_3 are the vector components of the quaternion $\mathbf{q}_{1:3}$ and q_4 is the scalar part. The constraint that must be maintained is the unit norm constraint. This vector has a magnitude of 1 which can be written as

$$\bar{\mathbf{q}} = \left|\mathbf{q}_{1:3}\right|^2 + q_4^2 = 1 \tag{3.56}$$

The unit norm of the quaternion must be equal to one in order for the attitude representation to remain orthogonal. Because the attitude system has three degrees-of-freedom (3-DOF), this four dimensional representation is not independent[54] and thus needs the constraint in Equation (3.56).

Quaternion multiplication is subject to special relations that must be preserved[53]. Using **i**, **j**, and **k** as the orthonormal basis of R^3, these products must be preserved for multiplication.

$$\begin{gathered}\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \\ \mathbf{ij} = \mathbf{k} = -\mathbf{ji} \\ \mathbf{jk} = \mathbf{i} = -\mathbf{kj} \\ \mathbf{ki} = \mathbf{j} = -\mathbf{ik}\end{gathered} \tag{3.57}$$

Multiplying two different quaternions **q** and **p** can be defined as

$$\bar{\mathbf{q}} \otimes \bar{\mathbf{p}} = \begin{bmatrix} p_4\mathbf{q}_{1:3} + q_4\mathbf{p}_{1:3} - \mathbf{q}_{1:3} \times \mathbf{p}_{1:3} \\ q_4 p_4 - \mathbf{q}_{1:3} \cdot \mathbf{p}_{1:3} \end{bmatrix} \tag{3.58}$$

The inverse of a quaternion can be written as

$$\bar{\mathbf{q}}^{-1} = \begin{bmatrix} -q_1 \\ -q_2 \\ -q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} -\mathbf{q}_{1:3} \\ q_4 \end{bmatrix} \tag{3.59}$$

Another important definition is that of the quaternion identity. This can be written as

$$\boldsymbol{q}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{3.60}$$

By keeping the unit norm constraint of the quaternion, the attitude matrix can be defined as

$$A(\boldsymbol{q}) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{3.61}$$

Quaternions are widely utilized in attitude estimation due to the fact that they avoid singularities present for eigenaxis rotation, their kinematics have linear relationships with angular velocities, and the attitude matrix components as shown in Equation (3.61) are algebraic[54]. One caveat for using quaternions in attitude estimation is that the unit norm constraint specified in Equation (3.56) must be enforced. There are several ways to approach this issue with the most common way being to use the quaternion error between the estimation and measurements and associated state transition matrix of the error covariance. It is common to parametrize this error and several parametrizations exist depending on the application and types of maneuvers being performed[54]. This is due to the presence of singularities inherent in the parametrization process. The Modified Rodrigues Parameters (MRP) are frequently used as they allow for rotations up to 360°[54,55]. The relationship between the MRP and quaternion are given by

$$\boldsymbol{p} = \frac{4\boldsymbol{q}_{1:3}}{1+q_4} \tag{3.62}$$

where $\boldsymbol{p}$ are the MRP. This parametrization is a *3x1* vector. Inversely, quaternions are related to the MRP through

$$\boldsymbol{q} = \frac{1}{1+\|\mathbf{p}\|^2}\begin{bmatrix} 2\mathbf{p} \\ 1-\|\mathbf{p}\|^2 \end{bmatrix} \tag{3.63}$$

Another parametric representation of quaternions is the Classical Rodriques parameters (CRP). These are related to the quaternion according to

$$\mathbf{g} = \frac{\mathbf{q}_{1:3}}{\mathrm{q}_{14}} \tag{3.64}$$

where **g** is the CRP. Unlike the MRP, the CRP parametrization is only valid for rotations up to 180°. However, applications involving small rotations certainly make using the CRP an appropriate choice. One major advantage for using CRP is that they are related to the rotation matrix by Cayley Transform which is given by

$$\mathrm{R} = \left(I + \mathrm{Q}(\mathbf{g})\right)^{-1}\left(I - \mathrm{Q}(\mathbf{g})\right) \tag{3.65}$$

where I is a *3x3* identity matrix and the operator Q(**g**) is defined as the skew symmetric matrix

$$\mathrm{Q}(\mathbf{g}) = \begin{bmatrix} 0 & -\mathrm{g}_3 & \mathrm{g}_2 \\ \mathrm{g}_3 & 0 & -\mathrm{g}_1 \\ -\mathrm{g}_2 & \mathrm{g}_1 & 0 \end{bmatrix} \tag{3.66}$$

Quaternions in this dissertation are used to represent the rotation from the LVLH frame to the inspector's body frame. Furthermore, quaternion kinematics are also utilized to express the rotation of the camera frame with respect to the target. The latter rotation is described in the following section and is a vital component of the measurement model in the estimation filter. Section 3.6 will describe the resulting rotation and translation of the camera frames around a *3*-dimensional point cloud.

3.6. Optimal Linear Attitude Estimator

Consider two frames A and B as seen in Figure **3-6**. Frame B is related to Frame A through the rotation matrix $R_{A/B}$ and translation $\mathbf{T}_{A/B}$. The relationship between the same set of points in both frames is given by

$$\mathbf{p}_{i/A} = R_{A/B}\mathbf{p}_{i/B} + \mathbf{T}_{A/B} \tag{3.67}$$

where $\mathbf{p}_{i/A}$ and $\mathbf{p}_{i/B}$ are the set of points relative to each respective frame. These points are obtained from the stereopsis process detailed in Section 3.2. The set of these points create a point cloud and are observed from different camera frames as the camera moves through space.

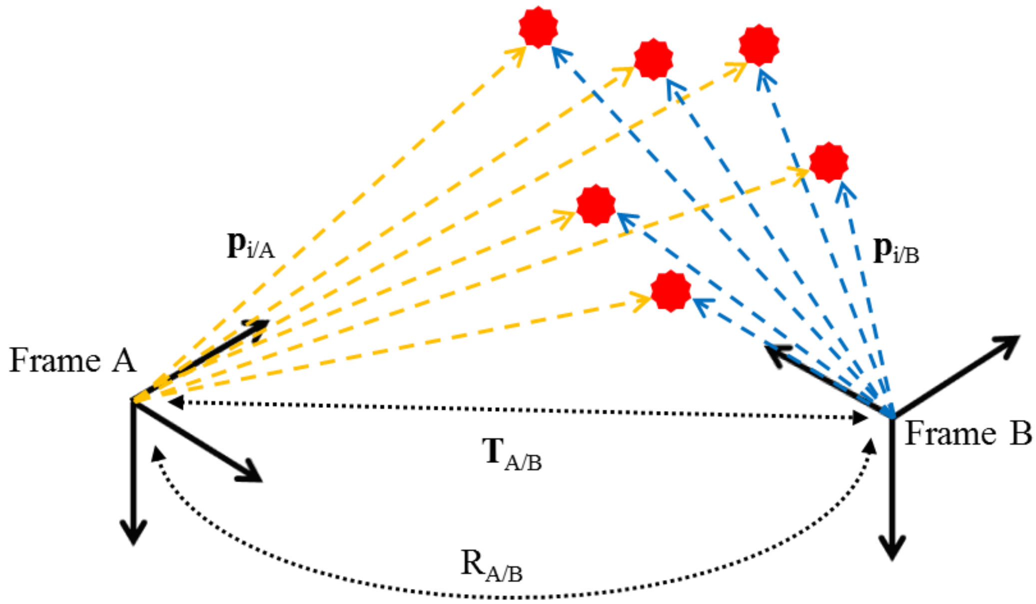

Figure **3-6**: Stereo reference Frames

In pose estimation, the goal is to determine the rotation and translation between subsequent frames. These points would be the landmark features in this scenario. To solve this problem, the centroid of the same set of points is computed by

$$\bar{\mathbf{p}}_{i/A} = \frac{1}{l}\sum_{k=1}^{l}\mathbf{p}_{k/A} \tag{3.68}$$

$$\bar{\mathbf{p}}_{i/B} = \frac{1}{l}\sum_{k=1}^{l}\mathbf{p}_{k/B} \tag{3.69}$$

where $\bar{\mathbf{p}}_{i/A}$ and $\bar{\mathbf{p}}_{i/B}$ are the centroids in each respective frame and l is the total number of points. Using the centroids and the set of points for each respective frame, the rotation matrix $R_{A/B}$ can be computed by

$$\left(\mathbf{p}_{i/B} - \bar{\mathbf{p}}_{i/B}\right) = R_{A/B}\left(\mathbf{p}_{i/A} - \bar{\mathbf{p}}_{i/A}\right) \tag{3.70}$$

Once the rotation matrix has been calculated, the translation can be solved for by substituting $R_{A/B}$ back into Equation (3.67).

The Optimal Linear Attitude Estimator (OLAE)[45,56,57] provides an accurate optimal method to determine the rotation and translation from two sets of point clouds. This approach solves for the rotation and translation without the need for any matrix inversion or decomposition. By taking the centroid relations defined in Equations (3.68)-(3.69), the difference between the centroid and any point in the point cloud is given by

$$\tilde{\mathbf{p}}_{i} = \mathbf{p}_{i} - \bar{\mathbf{p}} \tag{3.71}$$

Furthermore, the following parameters are defined to simplify further expressions:

$$\boldsymbol{\gamma} = \tilde{\mathbf{p}}_{i/B} - \tilde{\mathbf{p}}_{i/A} \tag{3.72}$$

$$\boldsymbol{\psi} = \tilde{\mathbf{p}}_{i/B} + \tilde{\mathbf{p}}_{i/A} \tag{3.73}$$

Equation (3.72) can be rewritten as

$$\boldsymbol{\gamma} = \left[\boldsymbol{\psi}\times\right]\mathbf{g} \tag{3.74}$$

Using the preceding equations and the requirement that the point cloud contains at least three points, the procedure to compute the CRP are as follows

$$\mathrm{B} = \sum_{i=1}^{N} \left[\boldsymbol{\psi}_{i}\times\right]^{T} \left[\boldsymbol{\psi}_{i}\times\right] \tag{3.75}$$

$$\mathbf{C} = \sum_{i=1}^{N} \left[\boldsymbol{\psi}_{i}\times\right]^{T} \boldsymbol{\gamma} \tag{3.76}$$

$$\mathbf{g} = \mathrm{B}^{-1}\mathbf{C} \tag{3.77}$$

The estimated CRP can be utilized along with the Cayley Transform to compute the estimated translation as

$$\mathbf{t} = \bar{\mathbf{p}}_{i/B} - R_{A/B}\bar{\mathbf{p}}_{i/A} \tag{3.78}$$

Equations (3.77)-(3.78) form the basis of the measurement equations for the localization filter in Section 3.18. The covariance of the attitude can be written as

$$P_{attitude} = -\left(1+\mathbf{g}^2\right)^{-1}\left(I-[\mathbf{g}\times]\right)M_m^{-1}\left(4M_m + D\right)M_m^{-1}\left(I-[\mathbf{g}\times]\right)^T \tag{3.79}$$

where the parameters M_m and D are defined below.

$$M_m = -4\sum_i \alpha_i \left(I - \mathbf{b}_i \mathbf{b}_i^T\right) \tag{3.80}$$

$$D = \sum_i \alpha_i \rho_i^2 \left(\rho_i \times \mathbf{g}\right)\left(\rho_i \times \mathbf{g}\right)^T \tag{3.81}$$

The matrix M_m is a real, symmetric, negative definite matrix and the parameters α and b are weights and the observed i^{th} unit vector. The covariance complies with Wahba optimality and provides approximation of the attitude error produced by OLAE.

3.7. Octrees

Shape estimation of the target spacecraft is created using an octree, which is a variant of a tree-based data structure in which each bin possesses eight octants. Three dimensional space is recursively partitioned with a center node within the bin. This recursive process occurs once a minimum bin size is reached. Each bin represents a cubic volume and is commonly referred to as a voxel. Higher resolution can be obtained by further subdivisions assuming the integrity of the data comprising the voxel space can be maintained. The denser the data comprising the octree, the more refined the octree can become. Since voxels represent a volume, they can model occupancy of the volume as long as that space contains information[58–60]. Figure **3-7** shows the hierarchal structure of an octree structure. Starting with a single bin, it is split into eight octants on the second level and three of those new bins are further split into another eight. The voxels shaded in black are occupied.

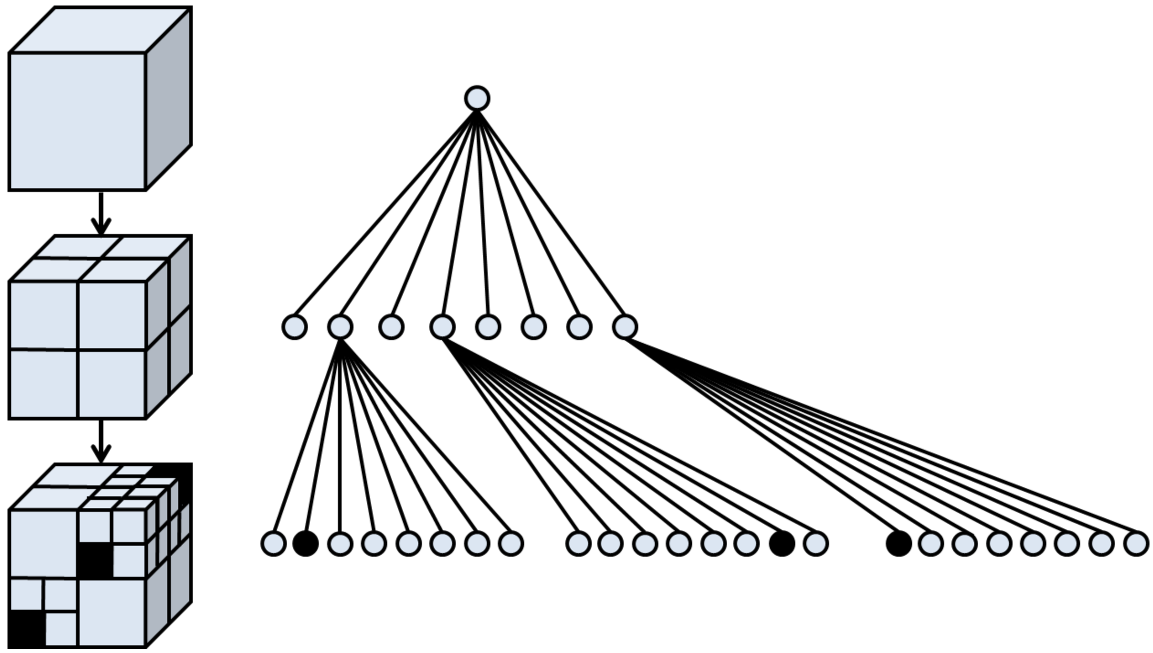

Figure **3-7**: Octree hierarchy

In robotic applications, it is appropriate to think of this occupancy in a probabilistic way. The higher the probability of occupancy of a volume, the certainty of the volume is better defined and has a lower associated covariance. This mapping represents the probability that the volume is occupied by the scene in which the observer is measuring[13]. In this work, the occupancy of the bins represents the probability that a target spacecraft is within this space. An advantage of using octrees is that the initialization of map volumes can wait until measurements are processed and smoothed by a filter[60]. As observations are conducted, the map is steadily built and does not require any priori information on the environment.

Uncertainty of the mapping can be quantified by using Shannon entropy[61]. In this expression, ρ_i represents the probability of occupancy and i denotes the ith bin.

$$U(t) = -\sum_{\mathrm{i}} \left[\rho_{\mathrm{i}} \ln(\rho_{\mathrm{i}}) + (1-\rho_{\mathrm{i}}) \ln(1-\rho_{\mathrm{i}}) \right] \tag{3.82}$$

The probability of occupancy is bounded between zero and one the associated uncertainty is smallest when the probability is equal to the lower or upper bound. One important caveat in evaluating the uncertainty and mapping is that points closest to the sensors are going to be biased due to the closer proximity. This is a common problem in stochastic mapping. To prevent over-bias in occupancy values, a saturation level is set such that voxels with overly dense information have their contribution limited to the map estimate and confidence of the octree.

Assuming a rigid target, it is possible to compute the centroid and principal geometric axes of the octree. The centroid of the octree can be computed as

$$\mathbf{r}_m = \frac{\sum_i \alpha_i \mathbf{r}_i}{\sum_i \alpha_i} \tag{3.83}$$

where $\mathbf{r}_m$ is the centroid, $\mathbf{r}_i$ is the coordinates of the bins, and is summed over i bins. To approximate the geometric principal axes, a principal component analysis (PCA) is conducted on the dataset comprising the bins. Essentially the principal axes are the eigenvectors of the dataset's covariance. Given a centroid position, a set of orthogonal axes are attached and extend in the directions in order of the three largest eigenvalues[62]. The covariance of the dataset is calculated according to

$$C = \frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{P}_i - \mathbf{r}_m\right)\left(\mathbf{P}_i - \mathbf{r}_m\right)^T \tag{3.84}$$

where $\mathbf{r}_m$ is the centroid computed from Equation (3.83), and points in the dataset $\mathbf{P}_i$. The eigenvalues of the covariance are found along with the associated eigenvectors. These eigenvectors define the principal axes of the dataset. Computing an octree of the SOLIDWORKS target model (refer to sections 3.18.2 and 4.1.3 for target model description) used in this dissertation and the associated centroid and geometric axes is shown in Figure **3-8(d)**.

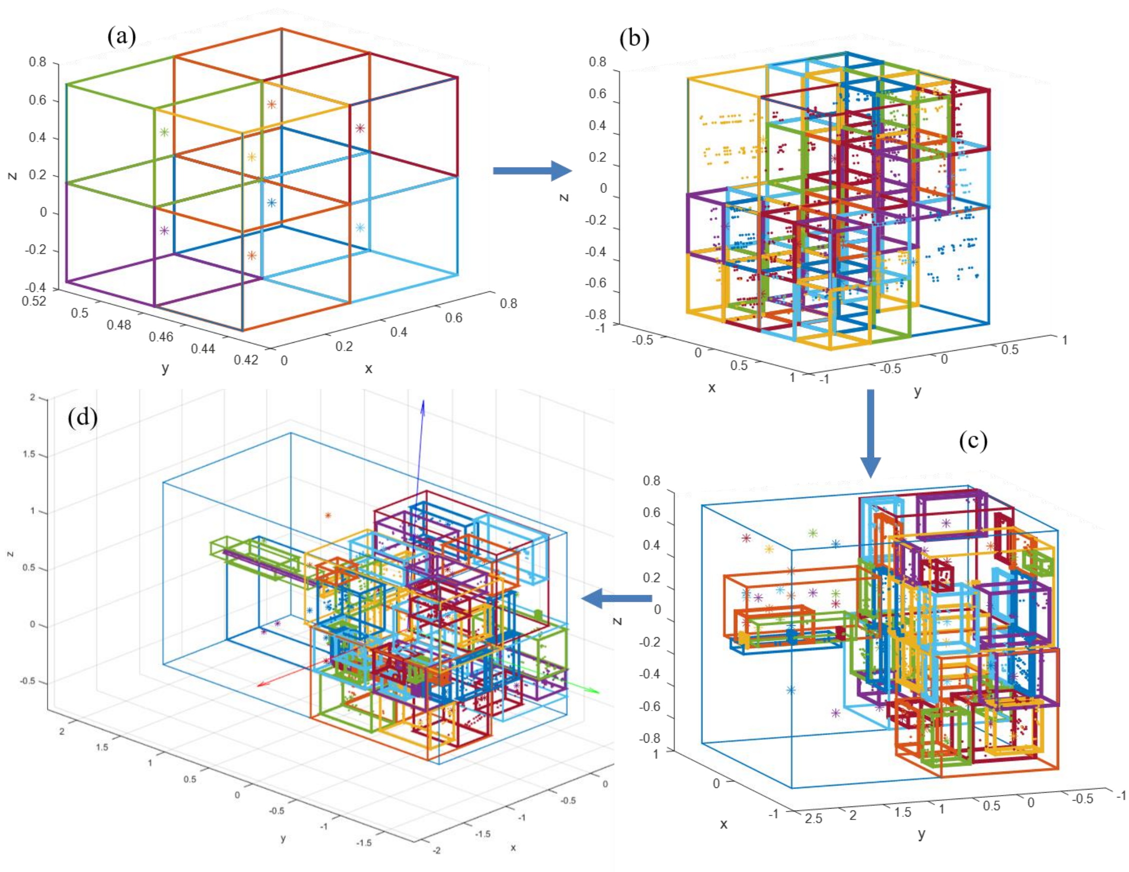

Figure **3-8**: Octree map progression with (a) three feature points of cubic body (b) full dataset of cubic body (c) cubic body with solar panel (d) full satellite model with geometric axes and geometric center estimate

Here, the bins of the octree are colored differently with the centroids (denoted by asterisks) and data points (denoted by dots) the same color as their parent bin. Figure **3-8** shows the progression of the octree map as it is recursively built. A maximum bin size limit was not set so that the bins are clearly visible. Otherwise, the bins would be too small to discern. In Figure **3-8(a)**, three of the eight bins are occupied with a single point forming a rough estimate of the actual cube size. Using the full dataset of the spacecraft cubic body leads to a better defined shape of the cubic spacecraft body in Figure **3-8(b)**. Several of the original eight bins were recursively subdivided to refine the shape. Next, a solar panel is added in addition to the cubic body in Figure **3-8(c)** and all empty bins are removed from the plot to make occupied bins more visible. In Figure **3-8(d)**, the full spacecraft model is utilized with geometric axes at the geometric center. Increased number of points in the point cloud increases the fidelity of the octree map. Setting a small maximum bin size will increase resolution of the shape with the bins more closely following the contours of the surface.

3.8. Random Sample and Consensus

The RANdom SAmple Consensus (RANSAC) is a commonly used method for determining the best data to fit a model[63–66]. Random subsets of data are taken from the whole data iteratively to find the best hypotheses of the data. Once a good hypothesis is found, these points are kept as the inliers and the rest are discarded as the outliers. There are two basic steps that occur iteratively in a general RANSAC algorithm: a hypothesize step in which minimal sample sets are randomly selected (hypotheses) from the entire dataset and the system model parameters are computed from this step. Next, a test is conducted to check which elements of the entire dataset are consistent with the system model parameters estimated in the first step[67]. These set of elements are referred to as the consensus set. The algorithm ends when the probability of finding a better consensus set drops below a pre-defined threshold.

A simple two-dimensional example to demonstrate the concept is finding the best line to fit a set of data. Figure **3-9** shows the iterative steps. Random hypothesis are taken from the raw data set. Model parameters are computed from this dataset which in this case are the slope and y-intercept of a line. Afterwards, the entire dataset is used to validate the model parameters computed from the random hypothesis by fitting a line based on the computed slope and y-intercept values. A distance function $\| y - mx - b < \varepsilon_{\text{Threshold}} \|_2$ (which is the distance between the data and the model) is applied to compare how the data set fits with the line and if it is within a threshold, the model is good.

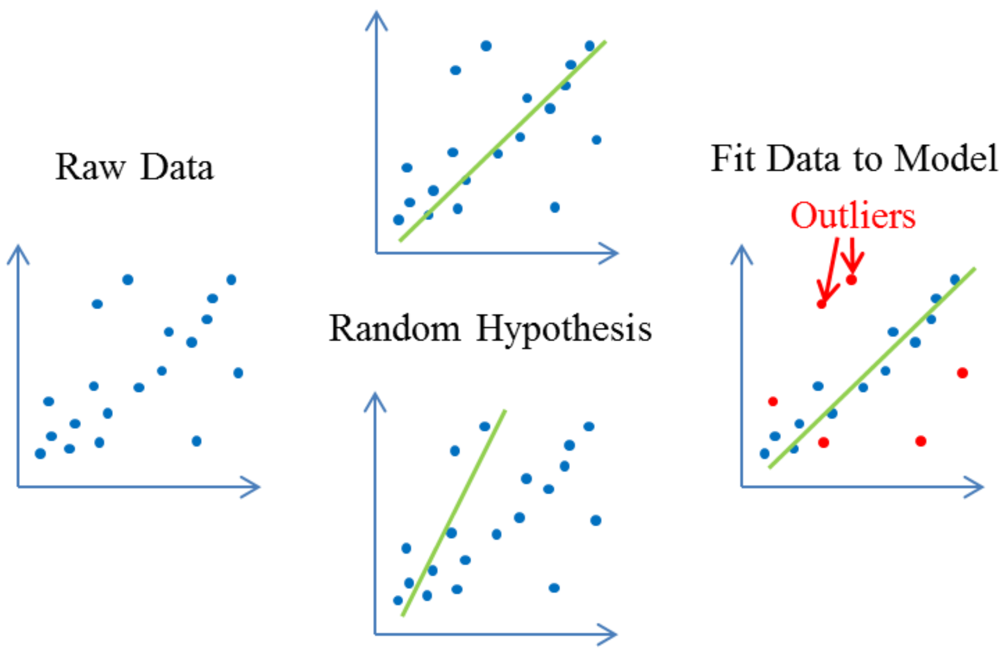

Figure **3-9**: Two-dimensional RANSAC example

Data points that cause degenerate model parameters are rejected as outliers and data points that produce consistent model parameters are kept as inliers.

In this dissertation, OLAE will be used as the model in which RANSAC will be looking for outliers. The model parameters of **g** and **t** will be used to compare random hypothesis of three-dimensional feature points. The distance function that will conduct the model consistency tests is

$$\| \mathbf{p}_{i/A} - R\mathbf{p}_{i/B} - \mathbf{t} < \varepsilon_{\text{Threshold}} \|_2 \tag{3.85}$$

where the rotation matrix R is a result of the Cayley Transform of the CRP.

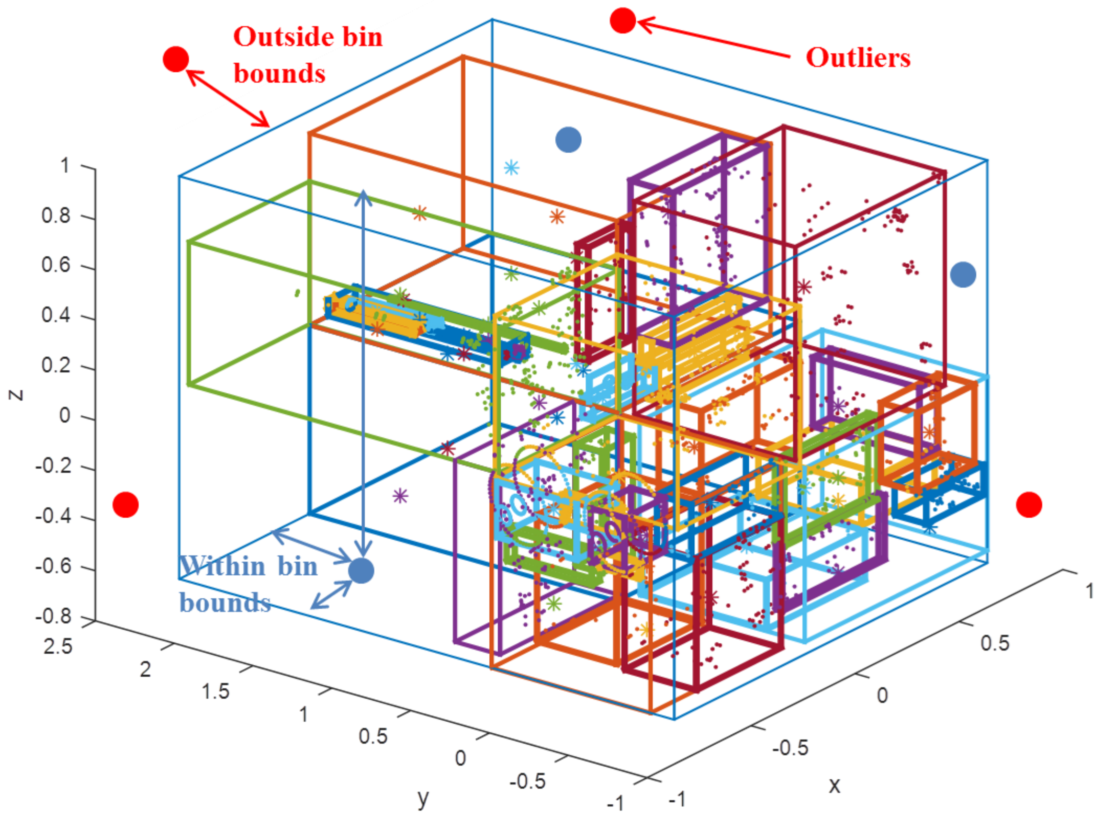

Figure **3-10**: Octree-based RANSAC

In addition to the RANSAC conducted using OLAE as its model, the octree mapping is also utilized to test for model inconsistency. Once the confidence in the mapping reaches a pre-defined threshold, the octree map of the target is used to check if the inliers are contained within the bin boundaries (Figure **3-10**). It is important to wait for the octree mapping to have low uncertainty before using it to prune bad measurements as a mapping with high uncertainty will potentially lead to outliers being accepted as inliers. Bad measurements can lead to a bad octree representation so care must be taken to avoid a feedback loop in which each continuously make each other worse. However, leveraging a well-defined map can greatly improve results from the RANSAC and reject inconsistent elements from the dataset.

3.9. Ray Tracing

The framework for a visual-based pose estimation problem is dependent upon having available images for the measurement system. Synthesizing images is accomplished through the development of scene generation architecture in MATLAB. This architecture is structured as follows: a CAD model is imported into MATLAB as a stereolithography (STL) file containing the vertices, facets, and normal. At any given time step, the vertices are projected onto the image plane via the projection equations outlined in Section 3.6. Once the image coordinates are obtained from the projection of the model vertices, a bounding box is defined around the region encompassing the projected points. This is done to reduce the number of calculations in the last step in which ray tracing is implemented to create the image.

The process of ray tracing is a straightforward concept. Figure **3-11** illustrates the process; given a coordinate in the world frame of a camera, a ray is cast from the camera origin through the middle of each pixel in a projected image plane towards an object. This ray will be referred to as the primary ray. Once the primary ray is computed from the camera origin to the projected pixel point, another calculation is conducted to determine if the ray intersects anything. The approach chosen for the scene generation framework uses the Möller-Trumbore algorithm[68]. This algorithm is a computationally efficient triangle intersection algorithm that exploits the barycentric coordinates of a triangle to determine if a line intersects it.

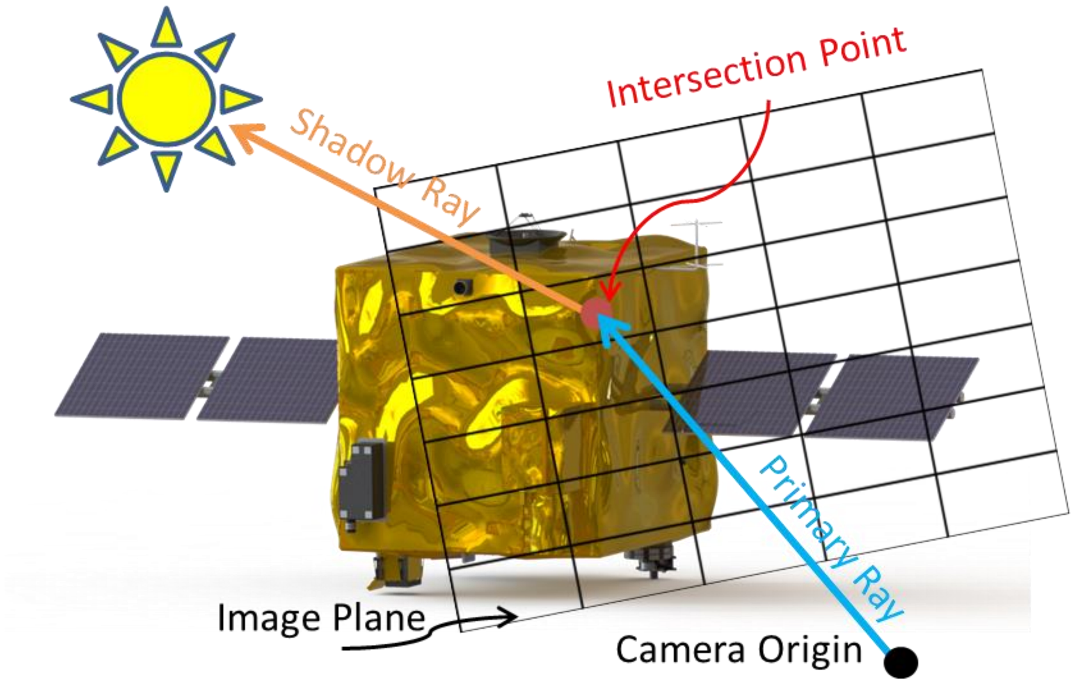

Figure **3-11**: Ray definition in ray tracing algorithm

For each primary ray drawn from the camera origin, every triangular mesh in the model must be checked to see if an intersection occurs. If multiple intersections occur (for example, the ray hits the front and backside of the object), the intersection closest to the camera origin is chosen. Once an intersection is determined, a secondary ray is drawn from this intersection point to the light source. Again, all triangular meshes are checked again for the purposes of determining if the intersection point is obstructed by another surface or has direct line of sight to the light source. This ray will be referred to as the shadow ray. After the shadow ray is computed, the color and brightness of the pixel is determined according the chosen lighting model. In this framework, the Phong Model[69] is used which is detailed in the following section. In the event that the primary ray intersects nothing, the algorithm passes onto the next pixel. However, all the primary ray calculations on the triangular mesh are still done for empty pixels.

Since this algorithm iterates over every pixel in the image, the computational requirements can be become extremely high. Computational speed is highly dependent on the number of model vertices and the resolution of the synthesized image. Aside from reducing the CAD model complexity and lowering the resolution, two other techniques were implemented to dramatically improve computational speed. Unless the camera origin is very close to the target object or zoomed in, the object does not encompass the entire image frame. As a result, lots of computations are wasted on empty space. This space can be eliminated entirely from computations by first projecting the vertices onto the image plane and drawing a bounding box around the area of interest.

Figure **3-12** shows a bounding box overlapping the portion of the image in which something is visible. It can be assumed that any area outside this region is empty and thus does not need to be computed in the ray tracing algorithm.

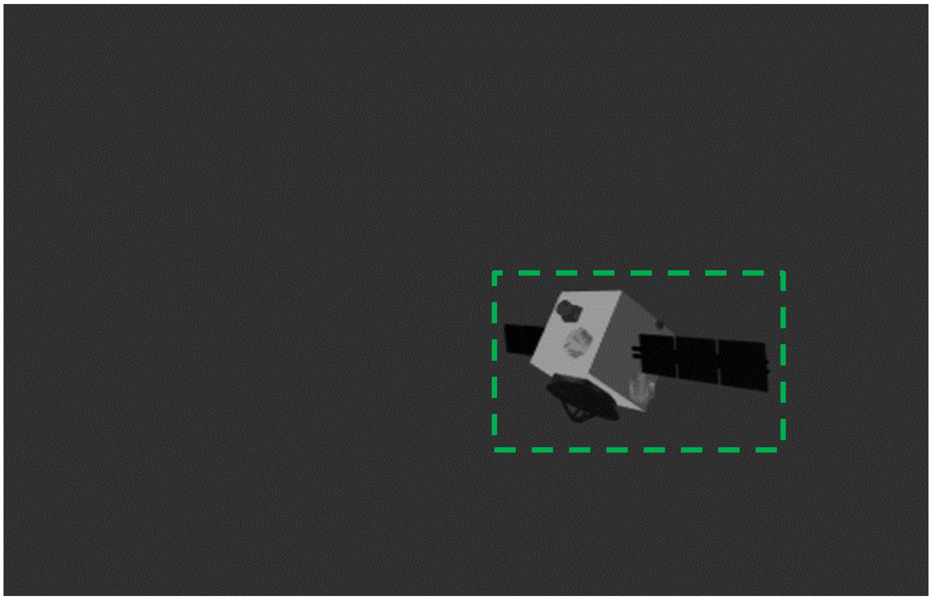

Figure **3-12**: Bounding box around projected vertices

This can eliminate many resulting calculations and dramatically improve the speed of the algorithm. An error tolerance is applied to the boundary region to slightly increase the bounding box size to ensure that edges are not clipped during the ray tracing procedure.

Finally, the second approach to improving the computational speed is by making use of parallel processing. Since the computation of a pixel is independent of the computation of other pixels, this process can be done in parallel. By using the Parallel Computing Toolbox in MATLAB, the ray tracing algorithm can be performed in parallel achieving further computational savings. An image produced by this process is shown in Figure **3-13**.

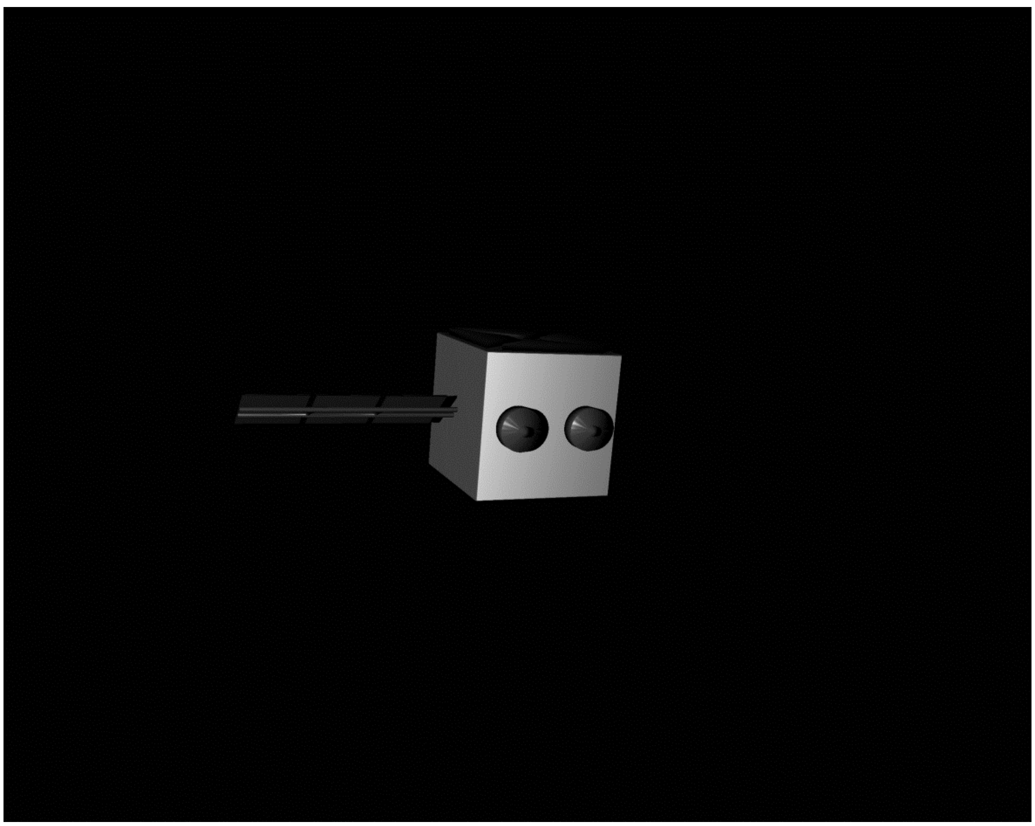

Figure **3-13**: Image produced by ray tracing algorithm

3.10. The Phong Model

Once a particular pixel is determined to be along a ray extending to the light source, the brightness and color on the grayscale needs to be computed for that pixel. Several approaches exist for this procedure ranging in accuracy and computational cost. For this work, speed was determined to be more important over accuracy as the ability to run many simulations would quickly get bottlenecked by a computationally intensive algorithm. While a method based on Bidirectional Reflectance Distribution Function (BRDF) can be very accurate, they tend to have higher computational requirements. The advantages of these methods are that it is physically accurate and energy conserving. For the purposes of this framework, the Phong Model[69,70] was chosen as the lighting model.

The Phong Model was very popular for many years due to its computational efficiency. However, the advent of faster multi-core processers, GPU computing and more physically accurate algorithms resulted in its relative disuse today. The Phong model works by making the assumption that the lighting on most materials can be approximated by a weighted sum of diffuse and specular components. Here, the diffuse term refers to the component of light that is invariant to the viewing angle. Diffuse materials can be modeled as reflecting light equally in all directions thus the brightness remains the same independent on the viewing orientation. The specular component behaves in an opposite manner; it is dependent on the viewing angle and can be considered akin to a glossy appearance. Diffuse surface color can be computed according to Equation (3.86) where ρ_s refers to the surface albedo, L_i is the amount of incident light energy, and φ is the angle between the normal vector and the light direction.

$$\text{Diffuse Color} = \frac{\rho_s}{\pi} L_i \cos(\varphi) \tag{3.86}$$

Specular color is computed using Equation (3.87) where $\boldsymbol{L}$ and $\mathbf{n}$ refer to the incident light direction and normal vector direction respectively; $\mathbf{V}$ is the viewing direction and e_s is referred to as the specular exponent. Combining these components and multiplying them by weights leads to the equation relating the specular and diffuse effects to the color at a pixel.

$$\text{Specular Color} = \left(\mathbf{V} \bullet \left(2\left(\mathbf{n} \bullet \boldsymbol{L}\right)\mathbf{n} - \boldsymbol{L}\right)\right)^{e_s} \tag{3.87}$$

In Equation (3.88), K_d refers to the diffuse weighting coefficient and K_S to the specular weighting coefficient.

$$\text{Pixel Color} = \text{Diffuse Color} \times K_d + \text{Specular Color} \times K_S \tag{3.88}$$

The drawback of this method is that it is not physically accurate. The weighting coefficients and the specular exponent do not have any physical meaning; adjusting these values is a trial and error to achieve the desired look of the image. Higher values of e_s result in a more focused specular highlight and the weighting confidents affect the overall intensity of the specular and diffuse components. However, the simple algebraic relationships enable a very computationally efficient algorithm. Since the focus of this dissertation is not on the computer vision aspect of generated images, accuracy is sacrificed for speed.

3.11. Multiplicative Extended Kalman Filter

As discussed earlier, quaternions need to have their unit norm constraint enforced in the estimation process. One such way to accomplish this is through the Multiplicative Extended Kalman Filter (MEKF)[55,71]. The fundamental premise of this algorithm is to propagate the four component quaternion state estimate by computing the quaternion error and its associated error covariance. An error quaternion is related to the estimate quaternion by

$$\delta\boldsymbol{q} = \boldsymbol{q} \otimes \hat{\boldsymbol{q}}^{-1} \tag{3.89}$$

where $\delta\boldsymbol{q}$ is the error quaternion, and $\boldsymbol{q}$ and $\hat{\boldsymbol{q}}$ are the measured and estimated quaternion respectively. The filter is propagated according to the quaternion kinematics which are defined as

$$\dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} \otimes \mathbf{q} = \frac{1}{2}\omega \otimes \mathbf{q} \tag{3.90}$$

where $\boldsymbol{\omega}$ is the angular velocity. Commonly, the angular velocity measurements are provided by gyros. These can provide updates to the state estimates. While the angular velocities are not directly estimated, the gyro biases are estimated. Assuming gyro measurements are available, they can be modeled by

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega} - \hat{\boldsymbol{\beta}} - \boldsymbol{\eta} \tag{3.91}$$

where $\hat{\boldsymbol{\omega}}$ and $\boldsymbol{\omega}$ are the estimated and measured angular velocities respectively, $\hat{\boldsymbol{\beta}}$ is the estimated gyro bias, and $\boldsymbol{\eta}$ zero-mean Gaussian white noise. Angular velocity is not directly estimated; rather the gyro bias is included in the state vector. The state vector is written as

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \mathbf{b} \end{bmatrix} \tag{3.92}$$

The associated error states are defined as

$$\delta\boldsymbol{x} = \begin{bmatrix} \delta\boldsymbol{q} \\ \delta\mathbf{b} \end{bmatrix} \tag{3.93}$$

where $\delta\boldsymbol{q}$ is the quaternion error as defined in Equation (3.93) and $\delta\boldsymbol{b}$ is the bias error. Bias error can be written as

$$\delta\boldsymbol{\beta} = \boldsymbol{\beta} - \hat{\boldsymbol{\beta}} \tag{3.94}$$

Once the parametrized error states and covariance are updated with the Kalman gain, the parametrized states must be converted back to the quaternion parametrization. This is done by

$$\mathbf{g} = \frac{\mathbf{q}_{1:3}}{q_{14}} \tag{3.95}$$

Finally, the quaternion estimate must be computed from the error quaternion given by

$$\hat{\boldsymbol{q}}^{+} = \delta\boldsymbol{q} \otimes \hat{\boldsymbol{q}}^{-} = \delta\mathbf{q}(\delta\mathbf{g}) \otimes \hat{\mathbf{q}}^{-} \tag{3.96}$$

For discrete time computations, the components of the error covariance can be computed by

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \tag{3.97}$$

$$\Phi_{11} = I_{3x3} - \frac{[\hat{\boldsymbol{\omega}}x]}{\|\hat{\boldsymbol{\omega}}\|}\sin(\|\hat{\boldsymbol{\omega}}\|\Delta t) + \frac{[\hat{\boldsymbol{\omega}}x]^2}{\|\hat{\boldsymbol{\omega}}\|^2}\{1-\cos(\|\hat{\boldsymbol{\omega}}\|\Delta t)\} \tag{3.98}$$

$$\Phi_{12} = I_{3x3}\Delta t + \frac{[\hat{\boldsymbol{\omega}}x]}{\|\hat{\boldsymbol{\omega}}\|^2}\{1-\cos(\|\hat{\boldsymbol{\omega}}\|\Delta t)\} - \frac{[\hat{\boldsymbol{\omega}}x]^2}{\|\hat{\boldsymbol{\omega}}\|^3}\{\|\hat{\boldsymbol{\omega}}\|\Delta t - \sin(\|\hat{\boldsymbol{\omega}}\|\Delta t)\} \tag{3.99}$$

$$\Phi_{21} = 0_{3x3} \tag{3.100}$$

$$\Phi_{22} = I_{3x3} \tag{3.101}$$

where Δt is the time period, $\hat{\boldsymbol{\omega}}x$ is the skew of $\boldsymbol{\omega}$, 0 is a *3 x 3* null matrix, and *I* is a *3 x 3* identity matrix[72]. The State Transition Matrix (STM) is denoted as Φ and is used to compute the covariance *P*. In summary, the algorithm can be seen in Figure **3-14**. The algorithm terminates once measurement updates cease. The MEKF is utilized in the inspector localization algorithm to handle the attitude and bias states for the inspector's attitude (LVLH to inspector body frame) and gyro bias respectively.

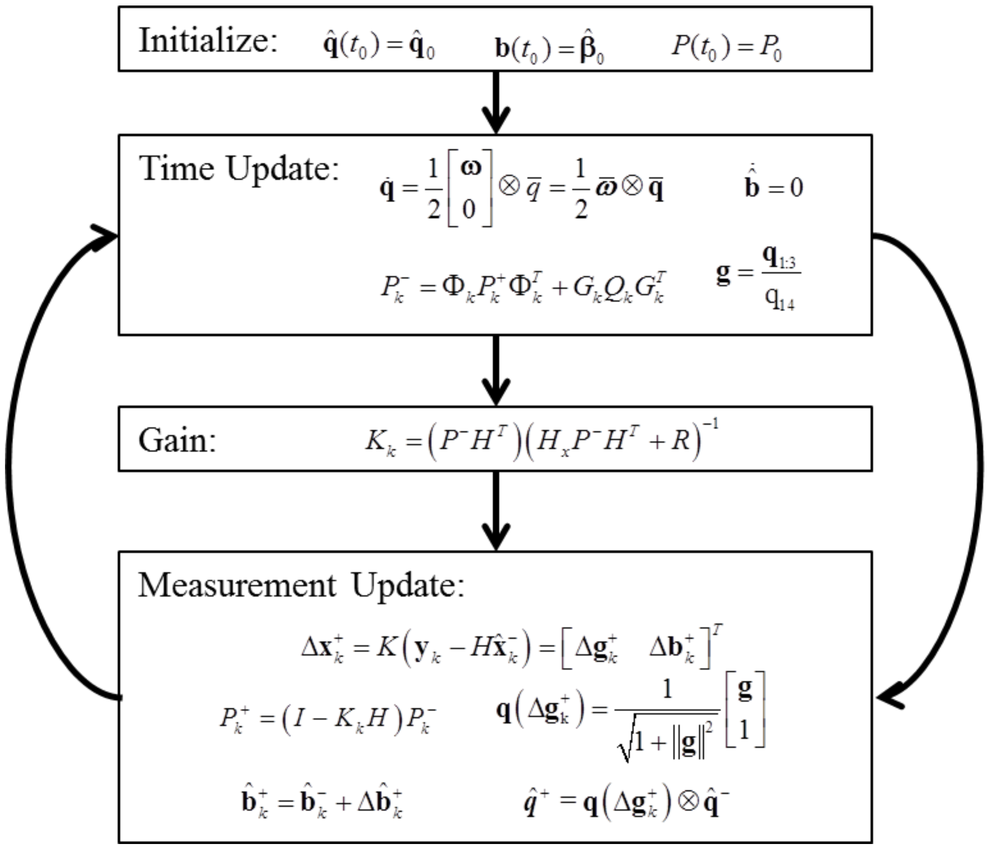

Figure **3-14**: Multiplicative Extended Kalman filter algorithm[71]

3.12. Consider Kalman Filter

One of the main contributions in this dissertation is to provide an online indirect mass estimation approach for thrust estimation. A coarse volume estimate is obtained from the octree map (Section 3.17) which possess an associated covariance. Estimating the volume is mildly observable whereas estimating mass is unobservable. Directly estimating either parameter in a filter could lead to divergence or an inherent bias. Furthermore, ignoring or assuming a presumed value for these parameters may also cause similar outcomes. Fortunately, there is an approach that can be utilized to consider the uncertainty associated with the volume estimate provided by the octree map while simultaneously excluding it from the estimation filter.

The Consider Kalman Filter, also referred to as the Schmidt Kalman Filter (SKF), was originally developed by Schmidt in the 1960s for handling the uncertainty pertaining to errors present in dynamic and measurement models[73]. Since the initial development of the SKF, there have been many improvements on the original algorithm[74–83]. The consider filter was adapted to both the Extended Kalman Filter (EKF)[76,77,83] and Unscented Kalman Filter (UKF)[78,79,81]. This dissertation uses the EKF formulation as it computationally less expensive than the UKF and will be referred to as a SEKF throughout the remainder of this work.

Adopting the formulation of the SEKF as seen in Woodbury[74] and Crassidis[54], the algorithm will be presented as follows. The discrete model for the states and measurement system can be written as

$$\mathbf{x}_{k+1} = \mathit{\Phi}_k \mathbf{x}_k + \mathit{\Theta}_k \mathbf{p} + \mathbf{w}_k \tag{3.102}$$

$$\mathbf{y}_k = H_{x_k} \mathbf{x}_k + H_{p_k} \mathbf{p} + \mathbf{v}_k \tag{3.103}$$

where $\mathit{\Phi}_k$ is the STM of the states and $\mathit{\Theta}_k$ is the STM of the consider parameters. Both **w** and **v** are assumed to be zero mean Gaussian random noise. The measurement matrices $H_{\mathrm{x_k}}$ and $H_{\mathrm{p_k}}$ are defined for both the estimated states (**x**) and the consider parameters (**p**), respectively.

In the propagation portion of the SEKF, the estimated states are updated at each time step. However, the consider parameter remains unchanged, as that is the underlying assumption of the consider filter[84].

$$\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k^+ + \Theta_k \hat{\mathbf{p}}_k^+ + w_k \tag{3.104}$$

$$\hat{\mathbf{p}}_{k+1}^- = \hat{\mathbf{p}}_k^+ \tag{3.105}$$

$$P_{xx_{k+1}}^- = \Phi_k P_{xx_k}^+ \Phi_k^T + \Phi_k P_{xp_k}^+ \Theta_k^T + \Theta_k P_{px_k}^+ \Phi_k^T + \Theta_k P_{pp_k} \Theta_k^T + Q_k \tag{3.106}$$

$$P^{-}_{xp_{k+1}} = \Phi_k P^{+}_{xp_k} + \Theta_k P_{pp_k} \tag{3.107}$$

As can be seen in Equation (3.105), the consider parameters take on the same estimate from the previous time step and remain unchanged throughout the algorithm. The covariance of the state estimates as seen in Equation (3.106) contains terms for both the states and the consider parameters. The covariance for the cross-correlation between states and parameters is defined by Equation (3.107). Consider parameter covariance $\mathrm{P_{pp_k}}$ also remains unchanged as the consider parameters. The optimal gain for the SEKF is similar to the classic Kalman gain except it contains extra terms encompassing the measurement matrix and covariances for the consider parameters.

$$K_k = \left(P^{-}_{xx}H^{T}_{x} + P^{-}_{xp}H^{T}_{p}\right)\left(H_x P^{-}_{xx}H^{T}_{x} + H_x P^{-}_{xp}H^{T}_{p} + H_p P^{-}_{px}H^{T}_{x} + H_p P^{-}_{pp}H^{T}_{p} + R\right)^{-1} \tag{3.108}$$

Terms denoted with subscript *x* refer to the state vector; terms denoted with subscript *p* refer to the consider parameters; terms with both *x* and *p* are cross-correlation components. Subscript k refers to the current time step. Finally, the estimated states and corresponding covariance are updated with the gain in the following formulations.

$$\hat{\mathbf{x}}^{+}_{k+1} = \hat{\mathbf{x}}^{-}_{k} + K_k\left(y_k - H_{x_k}\hat{\mathbf{x}}^{-}_{k} - H_{p_k}\hat{\mathbf{p}}_k\right) \tag{3.109}$$

$$\hat{\mathbf{p}}^{+}_{k} = \hat{\mathbf{p}}^{-}_{k} \tag{3.110}$$

$$P^{+}_{xx_k} = \left(I - K_k H_{x_k}\right)P^{-}_{xx_k} - K_k H_{p_k} P^{-}_{px_k} \tag{3.111}$$

$$P^{+}_{xp_k} = \left(I - K_k H_{x_k}\right)P^{-}_{xp_k} - K_k H_{p_k} P_{pp_k} \tag{3.112}$$

It should be noted that the consider parameter estimate remains unchanged through the end as shown in Equation (3.110). Equations (3.102)-(3.112) outline the common SEKF algorithm and are summarized in Figure **3-15**. Components colored in yellow are related to the consider parameters; consider parameters affect the state estimate dynamics and covariance while reaming unchanged itself. The consider parameters and associated covariance are colored in blue.

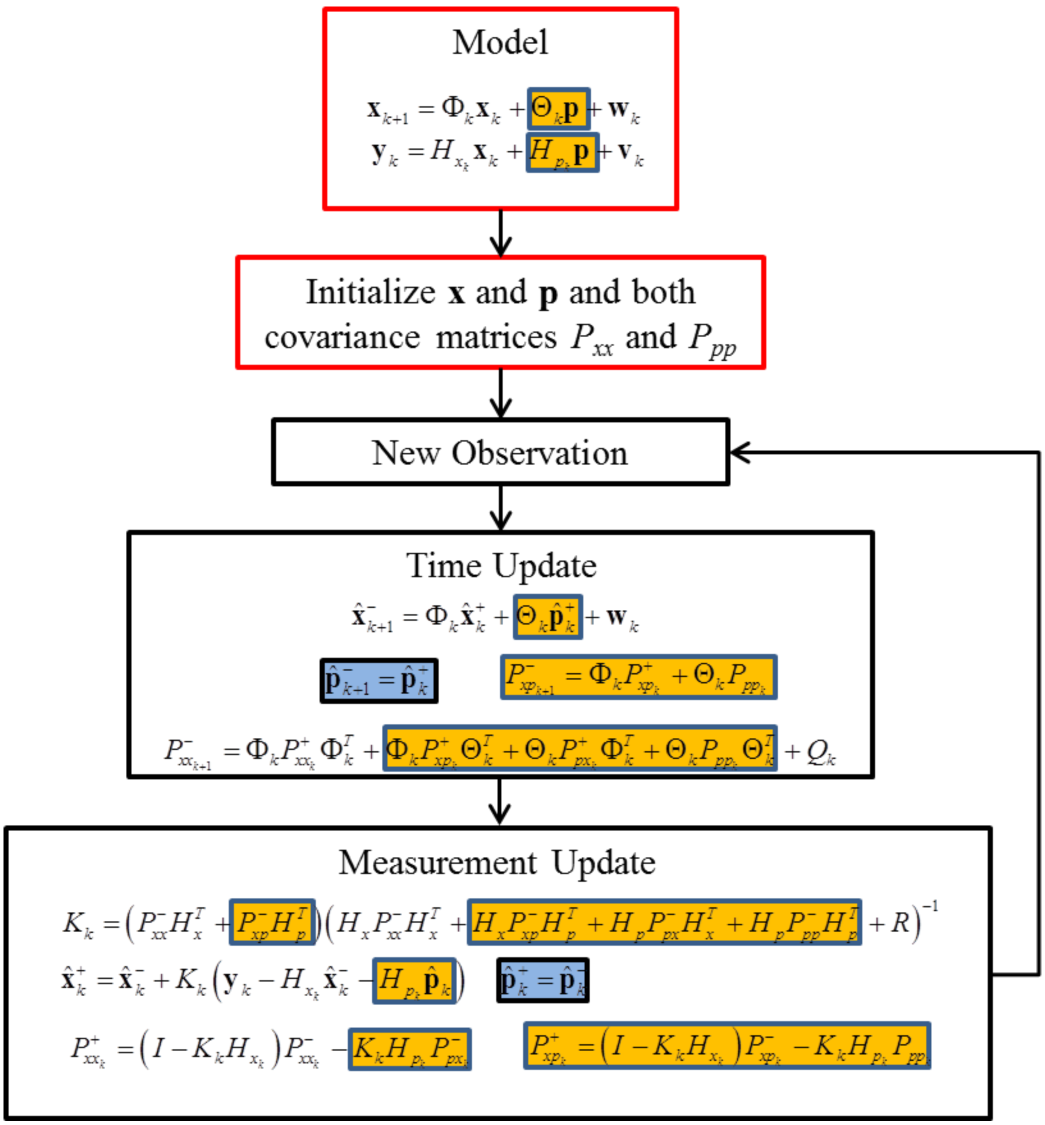

Figure **3-15**: SEKF algorithm[83]

3.13. Variable State Dimension Filter

Many approaches exist for detecting unknown maneuvers[35–40] with varying degrees of advantages and disadvantages. This dissertation follows the formulation of the VSD approach as outlined in Goff[35] and Bar-Shalom[36]. A summary of the VSD procedure is shown in Figure **3-16**. The VSD filter can be split in two models. During the time steps in which no maneuver is detected, the Kalman Filter runs normally; this portion of the algorithm is referred to as the quiescent model. The quiescent states to be estimated are the same as those covered in the section detailing the differential equations of motion. These include the relative position and velocity as shown:

$$\mathbf{x}_Q = \begin{bmatrix} x_I & x_J & x_K & v_I & v_J & v_K \end{bmatrix} \tag{3.113}$$

The accompanying state covariance is simply a 6x6 matrix corresponding to the position and velocity states. Once a maneuver is detected, the measurement residuals will begin to grow as the quiescent model runs. A metric called the Mahalanobis distance (ψ) is defined which utilizes residuals ($\mathbf{v}$), covariance, and measurement covariance R, to quantify the growing uncertainty in the estimates:

$$\Psi_k = \boldsymbol{v}_k^T \left(H_k P_k H_k^T + R_k \right)^{-1} \boldsymbol{v}_k \tag{3.114}$$

A predefined threshold is chosen such that when the Mahalanobis distance surpasses it, the quiescent model transitions to the VSD model. Under this paradigm, the quiescent model states are expanded to encompass the thrust states.

$$\mathbf{x}_{VSD} = \begin{bmatrix} x_I & x_J & x_K & v_I & v_J & v_K & a_{th_I} & a_{th_J} & a_{th_K} \end{bmatrix} \tag{3.115}$$

The thrust states are initialized as zeros. Additionally, the state covariance is expanded to accommodate the added thrust states as:

$$P_{VSD} = \begin{bmatrix} P_{6x6} & \vdots & 0_{6x3} \\ \cdots & \vdots & \cdots \\ 0_{3x6} & \vdots & P_{\tau_{3x3}} \end{bmatrix} \tag{3.116}$$

As researched by Goff [35], the covariance is inflated after the model switches to account for the uncertainty in the present state estimation as a result of the maneuvers. This is accomplished by increasing the values of the covariance as long as the trace of the covariance is less than a predefined threshold. The repeating calculation is performed according to the following heuristically determined relationship:

$$\hat{P}_k = 10\hat{P}_k \tag{3.117}$$

In Equation (3.117), the covariance is iteratively multiplied by *10* until a threshold is reached. Choice of multiplying the covariance by *10* was arbitrarily chosen by Goff [35] and could be tweaked. The threshold in which the operation stops is more important and directly affects filter performance. State estimation will continue within the VSD model until maneuver time ceases. Again, the Mahalanobis distance will become larger until it surpasses the threshold, which will cause the model to switch back to the quiescent states. The thrust states are removed from the state vector and covariance, and the same procedure of inflating the covariance occurs before continuing the estimation.

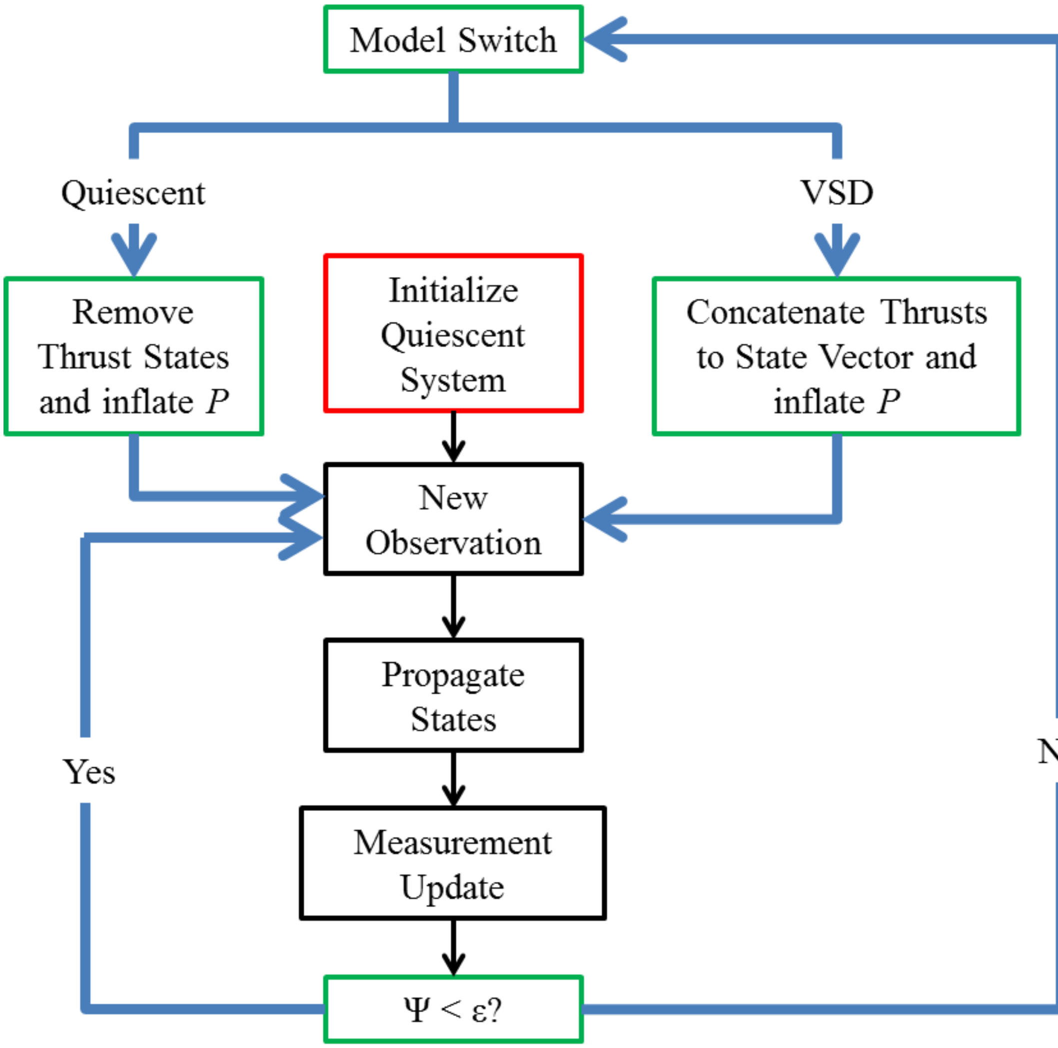

Figure **3-16**: Variable State Dimension Filter model switching[85]

3.14. The Consider Variable State Dimension Filter

The formulation of a VSD approach within the architecture of a SEKF was developed previously by Scarcella et al[85] as a foundational part of this dissertation. In the VSD approach outlined in the prior section, convergence of the thrust states are enabled through inflating the covariance to increase the uncertainty in the states post thrust maneuver. Furthermore, convergence is also helped through the addition of the process noise in the Kalman Filter at each time step. In both instances, the additive noise is subject to precisely tuning the covariance inflation threshold and the overall level of process noise. This can be partially mitigated through using the interacting multiple model (IMM), as each model uses a different level of process noise.

Inherently built into the consider filter design is the ability to handle uncertain biases within the model dynamics and measurements. The associated uncertainty from these consider parameters adds another, more targeted source of adjustment to the state covariance and thus the gain. Using these advantages with minimal extra computational power, this framework is applied to both the quiescent and VSD models of the maneuver detection scheme as shown in Figure **3-17**. The consider parameters in the quiescent model are present in both the quiescent and VSD portions of the algorithm. However, the consider parameters related to the thrusting component are present only in the VSD model.

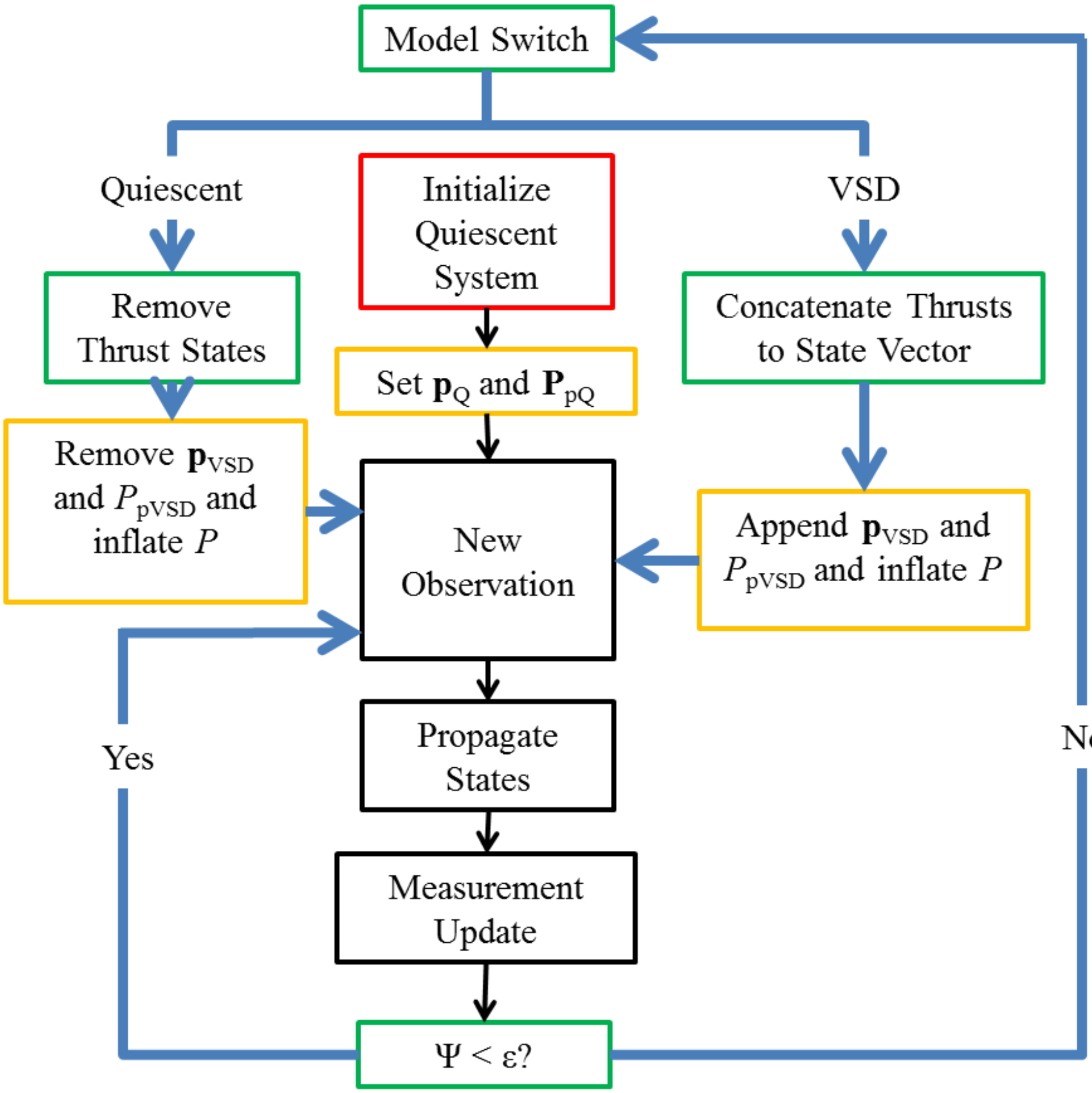

Figure **3-17**: Consider Variable State Dimension Filter model switching[85]

In the quiescent model, the algorithm is the same as described in the previous section on the consider Kalman filter. The only difference regarding the VSD formulation of the quiescent model is in the calculation of the Mahalanobis distance.

$$\Psi_k = v_k^T \left(H_x P_{xx}^- H_x^T + H_x P_{xp}^- H_p^T + H_p P_{px}^- H_x^T + H_p P_{pp}^- H_p^T + R \right)^{-1} v_k \qquad (3.118)$$

Once a maneuver is detected and the model switches accordingly, the thrust states are concatenated to the state vector as before. However, the consider parameters need to be accounted for. This can be more easily shown using the augmented state vector definition of the consider filter. In general, the state vector and covariance for a consider filter is defined as

$$\mathbf{z}_{Q_k}^{-} = \begin{bmatrix} x_Q \\ p_Q \end{bmatrix} \tag{3.119}$$

$$P_Q^{-} = \begin{bmatrix} P_x & P_{xp_Q} \\ P_{p_Q x} & P_{p_Q} \end{bmatrix} \tag{3.120}$$

where x are the estimated states, p are the estimated consider parameters, and subscript Q denotes the quiescent states. In the quiescent model of the VSD, this definition applies to the formulation. However, after transitioning to the maneuver detection portion of the algorithm, the quiescent consider parameter needs to be moved below the new thrust states since there cannot be a consider parameter in-between the estimated states. The augmented state vector then becomes

$$\mathbf{z}_{VSD_k}^{-} = \begin{bmatrix} x_k \\ x_{th_a k} \\ p_Q \\ p_{VSD} \end{bmatrix} \tag{3.121}$$

where the quiescent consider parameter is moved below the new thrust states and the VSD consider parameter is added to the end. The augmented covariance receives a similar adjustment wherein the cross-correlation terms in the quiescent state covariance are kept and moved to the appropriate positions as seen below

$$P_{VSD}^{-} = \begin{bmatrix} P_x & 0 & P_{xp_Q} & 0 \\ 0 & P_\tau & 0 & 0 \\ P_{p_Q x} & 0 & P_{p_Q} & 0 \\ 0 & 0 & 0 & P_{p_{VSD}} \end{bmatrix} \tag{3.122}$$

Keeping these terms retains the information between the states and consider parameters in the covariance during the model switch. Once the maneuver ends and the filter detects it, the thrust states and associated VSD consider parameter are removed and reduced back to Equation (3.119). The same thing occurs for the covariance; the elements of the covariance related to the thrust and VSD consider parameters are removed.

$$P_Q^{-} = \begin{bmatrix} P_x & \vdots & P_{xp_Q} & \vdots \\ \cdots & \vdots & \cdots & \vdots \\ P_{p_Q x} & \vdots & P_{p_Q} & \vdots \\ \cdots & \vdots & \cdots & \vdots \end{bmatrix} \Rightarrow P_Q^{-} = \begin{bmatrix} P_x & P_{xp_Q} \\ P_{p_Q x} & P_{p_Q} \end{bmatrix} \tag{3.123}$$

Again, the cross-correlation terms are kept for the next time step in the switched model.

3.15. Consider VSD Filter Results

Results of this approach as shown in Scarcella et al[85] demonstrated the computational savings and accuracy compared to more robust and computationally expensive algorithms. These results validate using the combined VSD SEKF approach over using an EKF. This framework is used as a component of the inspector algorithm in Section 3.18. In this section, the capabilities of this algorithm are demonstrated such that their use in the whole inspector algorithm is proven.

Initial conditions for a near circular LEO orbit are shown in Table **3-1**. A maneuver begins 4500 seconds into the simulation and lasts for 750 seconds. In this scenario, the thrust acceleration only occurs in the in-track direction with an acceleration of $1.875 \text{x} 10^{-4}$ m/s^2. A small thrust acceleration was chosen to determine the filter's effectiveness. Measurements are sampled at *1* Hz and include range and range rate with standard deviations of *1* m/s and *0.05* m/s^2, respectively. These are representative of typical sensor errors[35]. Tuning parameters such as the Mahalanobis distance threshold and the covariance inflation threshold are not optimally set. This is done to see how the filters handle a general situation in which the filter is not perfectly tuned.

There are two consider parameters set in this simulation; the quiescent consider parameter is the difference between the ballistic coefficients of both satellites (ΔBC) in the air drag acceleration in Equation (3.124). Mass of the deputy spacecraft is the second consider parameter. This parameter is solely used in the maneuver detection model of the algorithm. It is assumed that the propellant consumption from the maneuver does not have a noticeable effect on the total mass of the spacecraft since the duration is short. As a result, the nominal mass remains unchanged throughout the simulation.

$$a_d = \frac{1}{2}\left(BC_1 - BC_2\right)\rho v^2 = \frac{1}{2}\left(\Delta BC\right)\rho v^2 \tag{3.124}$$

$$\mathbf{p} = \begin{bmatrix} p_Q \\ p_{VSD} \end{bmatrix} = \begin{bmatrix} p_{\Delta BC} \\ p_m \end{bmatrix} = \begin{bmatrix} 2.5x10^{-4} \\ 800\,kg \end{bmatrix} \tag{3.125}$$

The estimated consider parameters are five percent larger than the nominal values. In the following series of simulations, an EKF and SEKF using a VSD filter are run using the same initial conditions and tuning parameters. Having set up the initial system, the simulation is propagated for roughly one orbital period. Two hundred Monte Carlo simulations were run to assess the overall performance of the SEKF relative to the EKF. In these runs, measurements were generated with additional noise with standard deviations as previously described. Furthermore, the initial conditions, covariance, and process noise had random noise applied in different runs. Both filters detect the maneuver about *200* seconds after the initial maneuver begins. Given the low thrust acceleration and sub-optimal tuning parameters, this behavior is expected. No inherent advantage is present in the detection of the maneuver using consider parameters. Table **3-2** shows RMS error for the quiescent states using a VSD filter. Results of the SEKF were on par with using the SEKF with a smoother without all the extra computational requirements.

Table **3-1**: Orbital Initial Conditions[85]

Initial Chief Orbit	Initial Relative Orbit
r = 8000 km	X = 500 m
i = 30°	Y = 0 m
Θ = 0°	Z = 86.6 m
Ω = 120°	$\dot{X}$ = 0 m/s
e = 0.001	$\dot{Y}$ = -0.71 m/s
h = 56,470 km^2/s	$\dot{Z}$ = 0 m/s

Table **3-2**: RMS error of quiescent states over 200 simulations[85]

	X (m)	*Y (m)*	*Z (m)*	Magnitude *(m)*	$\dot{X}$*(m/s)*	$\dot{Y}$*(m/s)*	$\dot{Z}$*(m/s)*	Magnitude *(m/s)*
SEKF	18.672	5.336	35.009	40.034	0.0321	0.0130	0.0324	0.0474
EKF	28.189	12.864	68.533	75.230	0.0384	0.0249	0.0872	0.0985
SEKF Smoother	15.539	5.657	19.278	25.539	0.0268	0.0090	0.0287	0.0403
EKF Smoother	25.949	5.746	21.010	33.879	0.0290	0.0092	0.0302	0.0408

Plotting of the residual error for the EKF and SEKF in Figure **3-18** shows that the EKF struggles to converge without the effects of the consider parameters. Both filters detect the maneuver about *200* seconds after the initial maneuver begins. Given the low thrust acceleration and sub-optimal tuning parameters, this behavior is expected. No inherent advantage is present in the detection of the maneuver using consider parameters. However, Figure **3-18** does show that after the maneuver occurs, the EKF (black line) struggles to remain within the 3σ bounds (dashed line). The SEKF after the initial covariance inflation procedure recovers within the duration of the simulation.

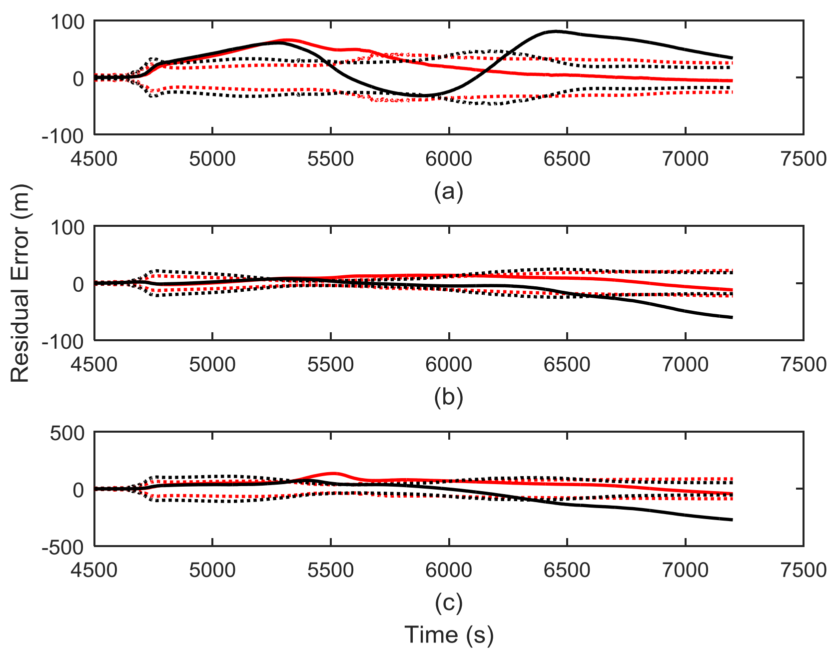

Figure **3-18**: Average residual error (solid line) with 3σ bounds (dashed line) of 200 simulations using the SEKF (red) and EKF (black) filters (a) in the radial position (b) in the in-track position (c) in the cross-track position[85]

Mahalanobis distances over all Monte Carlo simulations are plotted versus time in Figure **3-19**. The dashed line represents the threshold set for the model switching. If the Mahalanobis distance exceeds this threshold at any time step, a maneuver is either detected or ended depending on which model is currently active. One thing that becomes apparent is the frequency of false positives for both the SEKF and EKF before the initial maneuver at *4500* seconds. This further emphasizes that the initial tuning parameters were not optimal for the simulations; in this case, the threshold for the Mahalanobis distance was set slightly too low. Despite this, most maneuvers occur and end around the same time for both SEKF and EKF filters. However, the EKF has another large spike in the Mahalanobis distance around *6000* seconds. This indicates that a second false maneuver is potentially detected fairly frequently during the *200* Monte Carlo runs.

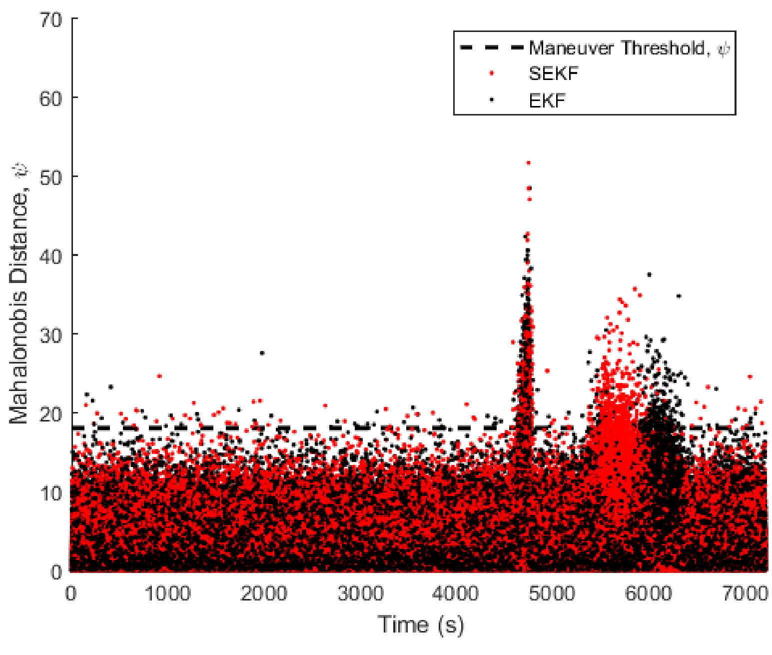

Figure **3-19**: Mahalanobis distance during Monte Carlo simulations[85]

The SEKF is less sensitive to the tuning parameters of the process noise and thresholds set for the system. With more tuning optimization, the EKF could demonstrate enhanced performance. Having a filter that can be more easily tuned (meaning less sensitive to sub-par optimal tuning parameters) and more robust to uncertainties in an actual environment is especially beneficial. Since there are thrust estimates following the end of the maneuver at *5250* seconds, this indicates that the end of the maneuver was not detected. Both filters struggle with detecting the end of the maneuver (Figure **3-20**) given the sub-optimal tuning parameters and low thrust accelerations.

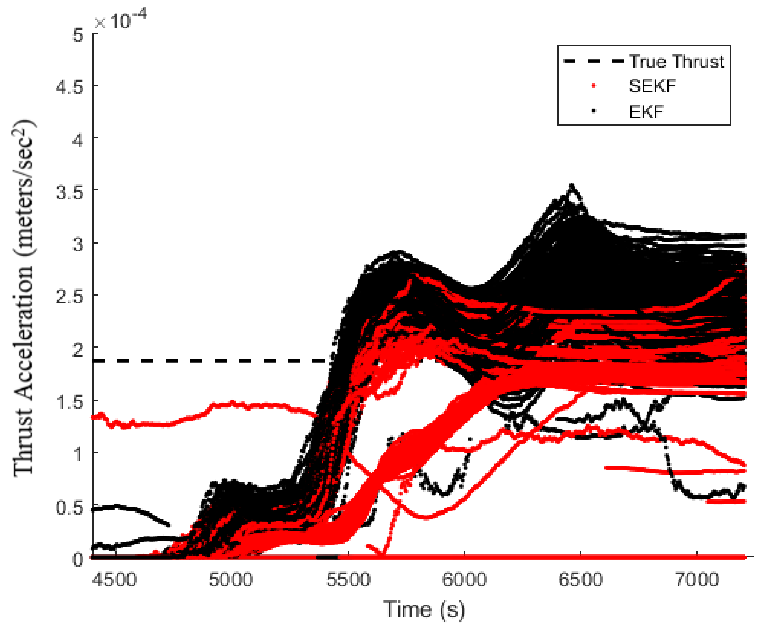

Figure **3-20**: Thrust acceleration magnitudes of SEKF and EKF filters[85]

Despite the poor thrust estimation, the SEKF under these difficult conditions still gives much enhanced trajectory estimate, as shown in Figure **3-21**. After the maneuver begins, the SEKF quickly recovers and converges back towards the true trajectory.

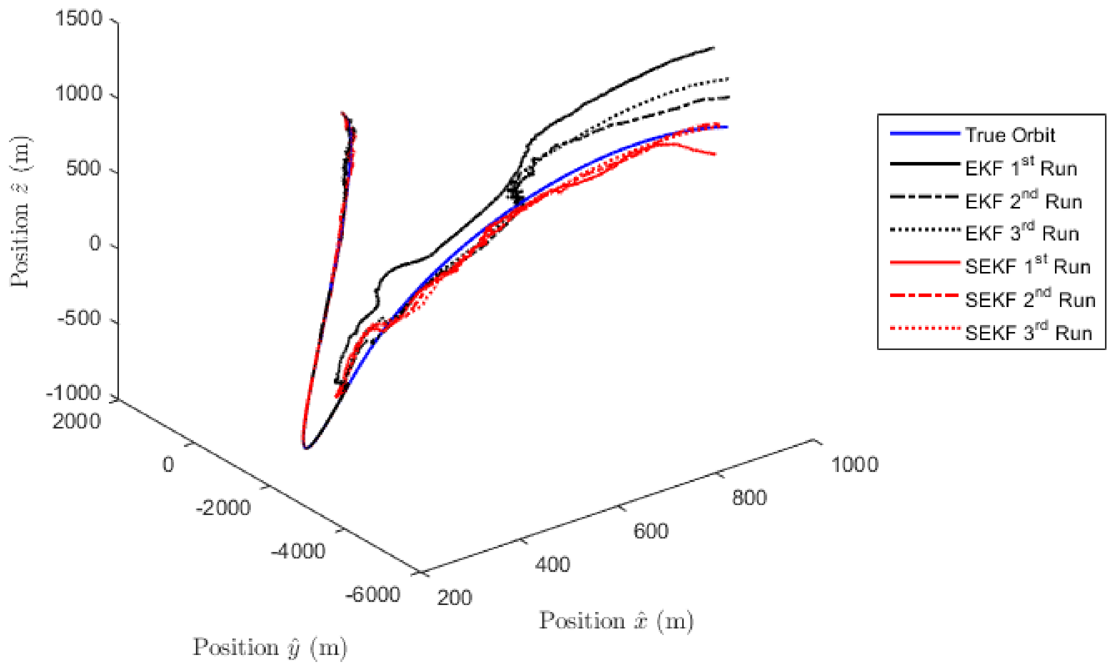

Figure **3-21**: Relative Orbit of Continuously Thrusting Spacecraft[85]

3.16. Interacting Multiple Model

An interacting multiple model filter has been used for object tracking[86,87] in previous studies. The algorithm works by generating *n* number of models in which the process noise is adjusted. Weights are assigned to these models and are initialized to be equal. As the filter propagates forward in time, the weights change based on how well a particular model performs; models that produce better state estimates will have its weight contribution increase. Varying the process noise helps to prevent filter divergence especially when estimating smaller thrusting maneuvers. Different levels of process noise provide the filter with the ability to find a better estimate of thrust as the maneuver is unknown and convergence can be difficult if the initial estimate guess is far from the truth. For low thrusting magnitudes, varying process noise provides enough variation such that the filter can find the model that works best.

The Interacting Multiple Model (IMM) approach proposed by Goff[35] only utilizes the IMM when switching to the maneuver detection portion of the VSD filter. Other methodologies use the IMM filter for both the quiescent and VSD model which is very inefficient for tracking. The IMM is in these approaches combines the quiescent and thrusting models at each step thus introducing errors when estimating a non-thrusting spacecraft using a thrust model. By only using the IMM in the thrusting model, these errors due to combining the models is avoided.

The algorithm begins by defining *n* number of models for the IMM. Higher number of models will increase the computational cost so caution needs to be used to use the lowest number of models needed to produce good state estimates. As mentioned before, the initial weights are initialized to be equal which will be used to weight each model in the filter to be combined for the final state and covariance estimate. This initial weight is defined as λ_i^k with initial states and covariance as $\hat{x}_i^k$ and $\hat{P}_i^k$ respectively. A probability matrix called $\mathrm{Pr}_{j|k}$ is defined containing the static mixing probabilities that switch from each model at every time step. The following steps are done initialize the weights, states, and covariance at the beginning of each filter time step. To determine the mixing weights, the sum of weights and probabilities for each model are done as follows

$$\bar{c}_k = \sum_{j=1}^{n} \mathrm{Pr}_{j|k}\, \lambda_{i-1}^{j} \tag{3.126}$$

The mixing weights are thus computed according to

$$\lambda_{i-1}^{j|k} = \frac{1}{\bar{c}_k} \mathrm{Pr}_{j|k}\, \lambda_{i-1}^{j} \tag{3.127}$$

Using these updated mixing weights, the mixed state and covariance for each model is calculated by

$$\bar{\mathbf{x}}_{i-1}^{j|k} = \sum_{j=1}^{n} \hat{\mathbf{x}}_{i-1}^{j} \lambda_{i-1}^{j|k} \tag{3.128}$$

$$\bar{\mathrm{P}}\mathrm{i}_{i-1}^{k} = \sum_{j=1}^{n} \lambda_{i-1}^{j|k} \left\{ \hat{\mathrm{P}}_{i-1}^{j} + \left[\hat{\mathbf{x}}_{i-1}^{j} - \tilde{\mathbf{x}}_{i-1}^{k} \right] \left[\hat{\mathbf{x}}_{i-1}^{j} - \tilde{\mathbf{x}}_{i-1}^{k} \right]^{T} \right\} \tag{3.129}$$

Initial states and covariance for each model are input into the filter and propagated. Normal steps for the EKF are conducted for each model state and covariance and stored. Furthermore, the residual and residual covariance (ν and S respectively) are stored for each model. Once the filter has *n* model number of states, covariance, residuals, and residual covariance, the weights are updated by

$$\Lambda_{i}^{k} = \frac{1}{\sqrt{\left|(2\pi)S_{i}^{k}\right|}} e^{-\frac{1}{2}(\nu_{i}^{k})^{T}(S_{i}^{k})^{-1}\nu_{i}^{k}} \tag{3.130}$$

$$\lambda_{i}^{k} = \frac{\Lambda_{i}^{k}\bar{c}_{k}}{\sum_{j=1}^{n} \Lambda_{i}^{j}\bar{c}_{j}} \tag{3.131}$$

All models are combined with their corresponding weights to determine the combined IMM estimate

$$\underline{\mathbf{x}}_{i} = \sum_{k=1}^{n} \hat{\mathbf{x}}_{i}^{k} \lambda_{i}^{k} \tag{3.132}$$

$$\underline{\mathrm{P}}_{i} = \sum_{k=1}^{n} \lambda_{i}^{k} \left\{ \hat{\mathrm{P}}_{i}^{k} + \left[\hat{\mathbf{x}}_{i}^{k} - \underline{\mathbf{x}}_{i} \right] \left[\hat{\mathbf{x}}_{i}^{k} - \underline{\mathbf{x}}_{i} \right]^{T} \right\} \tag{3.133}$$

These states and covariance are then used to initialize Equations (3.128) and (3.129) for the next time step. In this dissertation, target spacecraft density values comprise the models in the IMM. Further explanation how the model densities are chosen and utilized in the architecture are described in the following section.

3.17. Octree Target Mapping

Obtaining information about an unknown target is vital to rendezvous and proximity operations. By taking a series of stereo observations and storing the resulting feature points, an octree map is constructed. Since the feature points are not estimated as a part of the state vector in the filter from the increased computational cost, accuracy is sacrificed. Instead, feature points (obtained from the KLT algorithm) are updated to the current frame using the latest localization estimates (rotation and translation) and spacecraft body to camera frame rotation. Feature points are initially in the camera frame and require a transformation. This can lead to the map estimate smearing if too many bad observations are utilized to construct the map[13]. It is important to eliminate bad observations through RANSAC. Once the map reaches a pre-defined uncertainty threshold, the map itself is used to improve the feature point correspondence. Using the map before it reaches a high confidence can lead to a feedback loop in which bad observations make the map estimate worse leading to more bad observations making it through the RANSAC.

Building an accurate map serves multiple purposes shown in Figure **3-22**. The octree map is created in the target's body-fixed frame. The attitude of this frame is estimated using the principal component analysis procedure outlined in Section 3.7. Since the target is not rotating, better estimations of the principal axes cannot be obtained since they are not observable. Target geometric center (CoG) is also extrapolated from the octree map as described in Section 3.7. In the localization filter, the relative distance between the observer and target is assumed to be the distance between both centers of mass. The difference between the target's geometric center and center of mass is defined as $\mathbf{r}_{\text{offset}}$ (Figure **3-23**) and is estimated in the filter to correct the center of mass location. It also helps to better detect maneuvers by keeping track of this parameter.

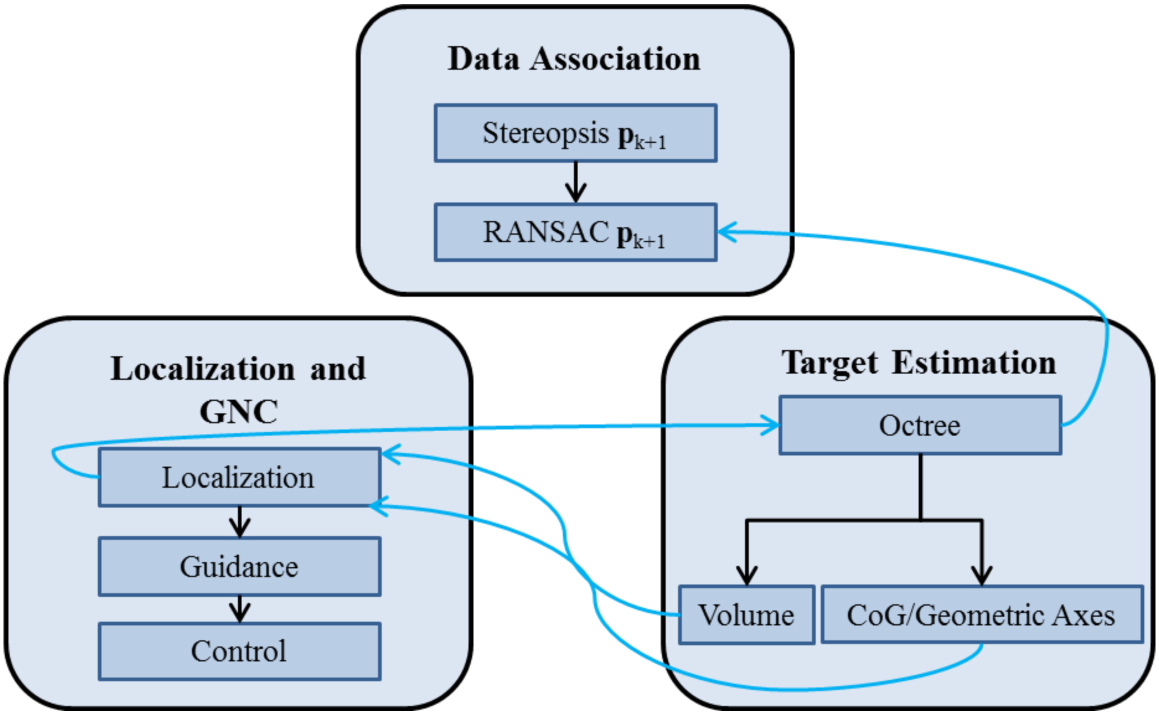

Figure **3-22**: Flow of octree estimates between RANSAC and localization

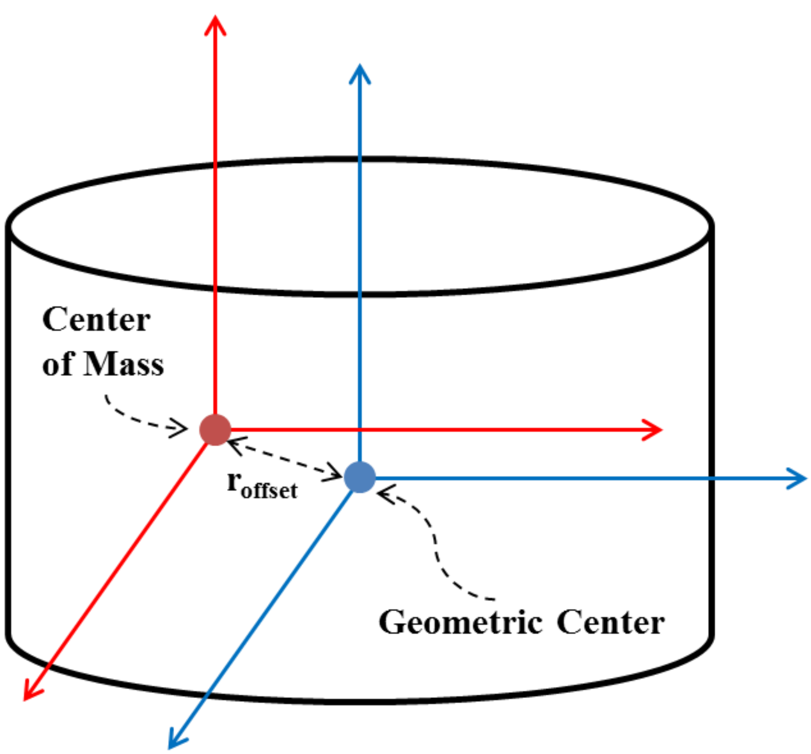

Figure **3-23**: Range offset between center of mass and geometric center

Most importantly, the volume is estimated from the octree map. Development of an online process to estimate mass for the maneuver detection scheme is a key component of this dissertation. An assumption is made that the target has a uniform density. Using publically available information on existing satellite dimensions and mass, a list of densities was computed assuming a uniform density distribution. These densities provide a bank of models to be used in accordance with the volume estimate and an IMM filter. Each density corresponds to a model in the IMM. A broad range of densities (Table **3-3**) are included due to the uncertainty in the target's density. The mixing weights corresponding to the models in the IMM will allow the filter to converge to the amalgamation of densities that best fits the observations.

Table **3-3**: Computed densities of existing spacecraft*

Density (*kg/m*^{3})
74.5578
153.8462
253.9062
150.7937
137.8378
192.0000
197.2387
107.9771

* Densities are computed from publically available information on existing satellites assuming a uniform density

Mass of the target is indirectly obtained from the volume estimate and the bank of densities. Estimation of the volume is done by computing the volume of each bin that is occupied. Equation (3.134) shows the computation of volume *v* as the summation of the length *l*, width *w*, times the height *h* of the i^{th} bin.

$$\mathrm{v} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{\alpha_{\mathrm{i}}}{\psi_{\mathrm{i}}} l_{\mathrm{i}} \mathrm{w}_{\mathrm{i}} \mathrm{h}_{\mathrm{i}} \tag{3.134}$$

Each bin is weighted by its occupancy value α and its saturation level ψ. Bins that are sparsely occupied account for less of the total volume than fully occupied bins. Furthermore, bins that are overly saturated are prevented from biasing the volume estimate. Bins that have no occupancy do not contribute to the total volume. The accuracy of the volume (as well as geometric center and principal axes) is highly dependent on the number of feature points comprising the octree and the bin size. Smaller maximum bin sizes produce better results. However, there is a trade-off between the length of bin size and computational cost. Smaller bin sizes produce more bins in the map and thus require more calculations for the octree itself, geometric center, and the principal axes.

Summarized in Table **3-4** are the maximum bin size allowed versus the volume estimate and computational time. All computations were done using *2000* feature points. The true volume obtained from the CAD model of the spacecraft is *1.8864* m^3. A maximum bin size of $3\ m^3$ has a very low computational cost but is over three times the true volume. Conversely, a bin size of $0.25\ m^3$ has a computational time roughly *137* times higher but only a *10.3*% difference from the true volume. Since the computation of the map is done every time an observation update occurs, a compromise must be made to keep computational times amenable for onboard spacecraft implementation. The estimation filter must utilize less accurate volume estimates for the sake of computational efficiency. Filter performance is judged on its ability to cope with these coarse volume estimates. This is why implementation of the IMM is required to overcome the effects of a coarse volume estimate. Details of this procedure are outlined in Section 3.18.

Table **3-4**: Effect of maximum bin size on volume accuracy and computational cost

Maximum Bin Size (*m*)	**Volume Estimate (m^3)**	**Computational Time (*sec*)**†
3.00	5.9450	0.009125
1.50	4.2886	0.086805
1.25	4.1993	0.086943
1.00	3.6438	0.094156
0.75	3.2994	0.584452
0.60	2.2692	0.690648
0.25	2.0824	1.249712

3.18. Inspector Localization Algorithm

The architecture of the inspector localization algorithm is one of the main contributions to this body of work. The full algorithm is presented and discussed including all assumptions and decisions for the approaches. This section includes the state dynamics, observation model, image processing, data association, implementation of octree mapping properties, and the guidance and control laws.

† Simulation done on an Intel Core i7-4770K CPU @ 3.50 GHz

3.18.1. Nonlinear State Dynamics

The purpose of the localization algorithm is to smooth out the rotation and translation estimates from OLAE. Utilizing the estimated target properties extrapolated from the octree model, the inspector localization also detects and estimates thrusting maneuvers. As described in Sections 3.13 and 3.14 on the VSD and Consider VSD filters, the maneuver detection algorithm consists of two components. A quiescent model describing the initial model states are initialized as

$$\hat{\mathbf{X}} = \begin{bmatrix} \mathbf{r}_{\mathrm{LVLH}} & \mathbf{v}_{\mathrm{LVLH}} & \mathbf{q}_{\mathrm{LVLH/S}} & \mathbf{b}_{\mathrm{gyro}} & \mathbf{r}_{\mathrm{offset}} \end{bmatrix} \tag{3.135}$$

where $\mathbf{r}_{\mathrm{LVLH}}$ is the relative distance between observer and target centers of mass in the LVLH frame; $\mathbf{v}_{\mathrm{LVLH}}$ is the relative velocity in the LVLH frame; $\mathbf{q}_{\mathrm{LVLH|S}}$ is the quaternion expressing the rotation from the LVLH frame to the observer body frame *S*; $\mathbf{b}_{\mathrm{gyro}}$ and $\mathbf{r}_{\mathrm{offset}}$ are the gyro bias states and offset from geometric center and center of mass (Figure **3-23**) respectively. The quiescent model states do not include the thrust states.

Once the Mahalanobis distance reaches a pre-defined threshold due to rising residuals, the model switches to a thrust state model. The state vector in Equation (3.135) is concatenated to include the thrust states $\mathbf{a}_{\mathrm{th}}$.

$$\hat{\mathbf{X}} = \begin{bmatrix} \mathbf{r}_{\mathrm{LVLH}} & \mathbf{v}_{\mathrm{LVLH}} & \mathbf{q}_{\mathrm{LVLH/S}} & \mathbf{b}_{\mathrm{gyro}} & \mathbf{r}_{\mathrm{offset}} & \mathbf{a}_{\mathrm{th}} \end{bmatrix} \tag{3.136}$$

$$\dot{\hat{\mathbf{b}}} = 0 \tag{3.137}$$

$$\mathbf{r}_{\mathrm{offset}} = 0 \tag{3.138}$$

Equations (3.48)-(3.50) are the dynamics governing $\mathbf{r}_{\mathrm{LVLH}}$ and $\mathbf{v}_{\mathrm{LVLH}}$ in the LVLH frame. Quaternion kinematics in Equation (3.108) govern the attitude state of the inspector $\mathbf{q}_{\mathrm{LVLH|S}}$. It is assumed that the bias states and the offset between the geometric center and center of mass do not change with respect to time. The external forces (F_{jx}, F_{jy}, and F_{jz}) in the quiescent model are nonexistent however, in the thrust model thrust states are added as

$$\mathbf{a}_{\mathrm{th}} = \frac{\mathbf{T}}{\mathrm{m}} \tag{3.139}$$

where m is the mass of the target spacecraft and $\mathbf{a}_{\mathrm{th}}$ is the acceleration due to thrust $\mathbf{T}$.

The attitude component of the state vector is done through a MEKF in which the quaternion states are parametrized. In this dissertation, the quaternions are parametrized by the CRP. The reason for this is the ability to obtain the rotation matrix from the Cayley Transformation upon which the OLAE is built upon. Within the inspector localization filter, the quiescent and thrust model states become

$$\hat{\mathbf{X}}^{*} = \begin{bmatrix} \mathbf{r}_{\mathrm{LVLH}} & \mathbf{v}_{\mathrm{LVLH}} & \mathbf{g}_{\mathrm{LVLH/S}} & \mathbf{b}_{\mathrm{gyro}} & \mathbf{r}_{\mathrm{offset}} \end{bmatrix} \tag{3.140}$$

$$\hat{\mathbf{X}}^{*} = \begin{bmatrix} \mathbf{r}_{\mathrm{LVLH}} & \mathbf{v}_{\mathrm{LVLH}} & \mathbf{g}_{\mathrm{LVLH/S}} & \mathbf{b}_{\mathrm{gyro}} & \mathbf{r}_{\mathrm{offset}} & \mathbf{a}_{\mathrm{th}} \end{bmatrix} \tag{3.141}$$

where the quaternion state has been replaced by $\mathbf{g}_{\mathrm{LVLH|S}}$. These reduced state vectors are used locally inside the filter and are used to update the full quaternion state as described in Section 3.8.

3.18.2. Observation Model

Observations made by the inspector are conducted via two stereo cameras. Each camera is assumed to be perfectly aligned thus making the stereo perspective equations in Section 3.2 valid. Both cameras are defined in the inspector spacecraft body frame S and the rotation from the body frame S and the camera frame C is defined as $R_{\mathrm{S/C}}$. This rotation is assumed to be known. These frames are defined in Figure **3-24** in which the spacecraft body frame $\mathbf{X}_{\mathrm{S}}$ axis points in the forward facing direction of the spacecraft. The $\mathbf{Y}_{\mathrm{S}}$ axis points in the opposite direction of the solar array and the $\mathbf{Z}_{\mathrm{S}}$ axis completes the triad. In the camera frame, the $\mathbf{Z}_{\mathrm{C}}$ axis points along the boresight of the camera in the same direction as $\mathbf{X}_S$ and $\mathbf{X}_C$ points in the same directions as the solar array. The $\mathbf{Y}_C$ axis completes the triad.

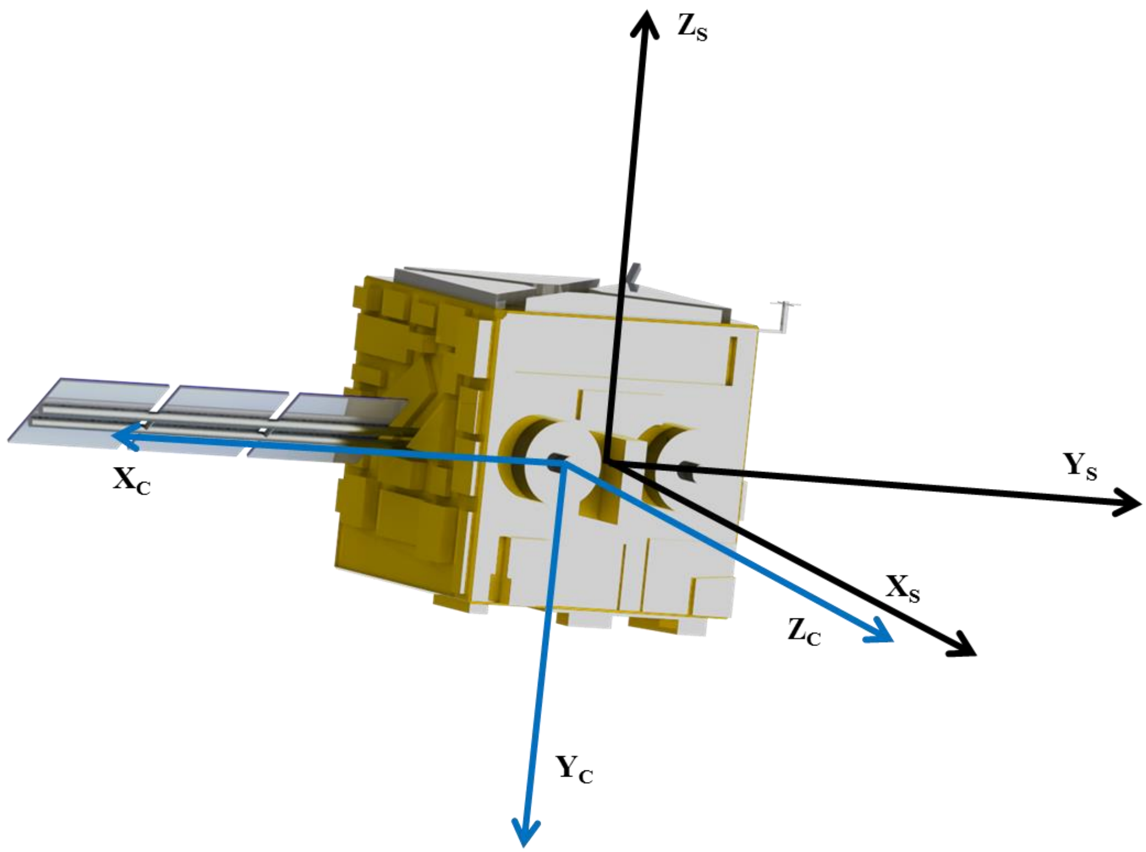

Figure **3-24**: Spacecraft body and camera frames

For any given observation time step, two images are generated for the left and right cameras respectively. The image generation procedure described in Sections 3.9 and 3.10 takes the above spacecraft CAD model and imports it into MATLAB as a STL file. Using the ray tracing procedure, greyscale images are generated based on both cameras' pose relative to the target CAD model. This procedure is depicted in Figure **3-25**. These images form the basis of observations for the inspector spacecraft.

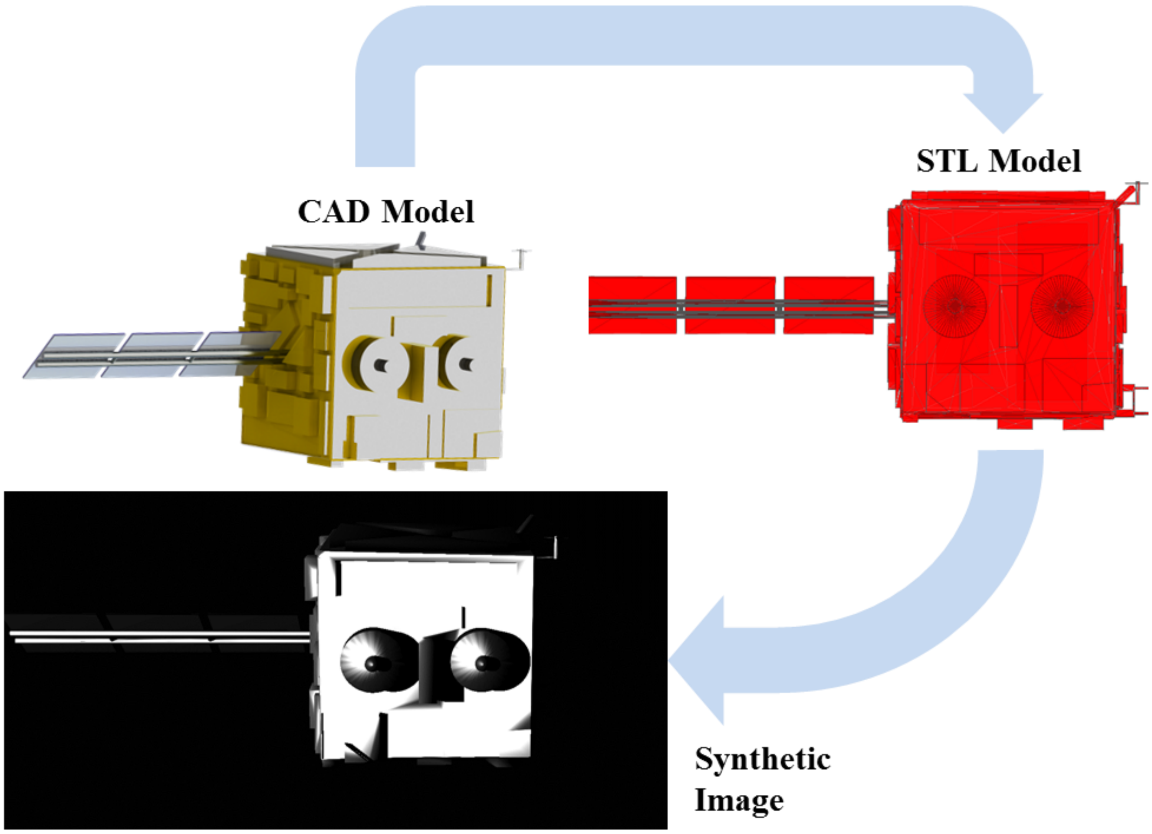

Figure **3-25**: The original CAD Model is imported as a STL and transformed into an image

Features from the left and right images are matched and triangulated to extrapolate three-dimensional feature points in the camera frame. When an observation occurs at some time step later ($k+1$), the same procedure is repeated. Next, features from the current and previous frame are matched and the resulting 3-dimensional feature pairs are run through RANSAC using OLAE as the model in which the set of feature points are compared. Once the outliers are removed, the best OLAE estimates of the rotation and translation between frames are used as the observation model for the inspector localization filter.

OLAE was chosen as the observation model for two main reasons. The more commonly used observation models for visual-based observations are based on pixel-level features[21,23,88–97].

However, performance is strongly related to the quality of images and lighting conditions. Over-saturation, occlusion, shadows, and other lighting conditions may affect the ability for a pixel feature based observation system to provide adequate information for the estimation filter. Furthermore, the model itself is highly non-linear which can introduce errors in the measurement update of an EKF. OLAE provides an approach that is computationally efficient and is linear in nature thus making it conducive for implementation in a filter-based estimation method. It is also more invariant to lighting conditions as the pixel features themselves are not used as the observation model; rather the movement between point clouds is the basis of the model. The rotation and translation are less dependent on perfect lighting conditions.

The observation model is thus defined as

$$\mathbf{y} = \begin{bmatrix} \mathbf{t}_{k|k+1} \\ \mathbf{g}_{k|k+1} \end{bmatrix} \tag{3.142}$$

where $\mathbf{t}_{k|k+1}$ is the translation between frame *k* and frame *k+1*; and $\mathbf{g}_{k|k+1}$ is the CRP representation of the rotation between frames *k* and *k+1*. These frames are from the corresponding observation estimates are thus

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{\mathbf{t}}_{k|k+1} \\ \hat{\mathbf{g}}_{k|k+1} \end{bmatrix} \tag{3.143}$$

The full relationship of the translation between frames *k* and frame *k+1* is written as

$$\hat{\mathbf{t}}_{k|k+1} = \mathrm{R}_{S/C}\mathrm{R}(\hat{\mathbf{g}}_{k+1})_{LVLH/S}\left(\mathbf{r}_{LVLH|k} - \mathbf{r}_{LVLH|k+1}\right) \tag{3.144}$$

where $\mathrm{R}(\hat{\mathbf{g}}_{k+1})$ is the Cayley Transform of the CRP state estimate $\mathbf{q}_{LVLH|S}$, and $\mathbf{r}_{LVLH|k}$ and $\mathbf{r}_{LVLH|k+1}$ are the previous and current state estimates of the inspector position relative to the target. The resulting translation from frame *k* to *k+1* is in the camera frame *C* relative to the LVLH frame. Derivation of the CRP measurement estimate ($\hat{\mathbf{g}}_{k|k+1}$) between frames starts with a transformation of the current and previous CRP state estimate into the camera frame.

$$\mathbf{g}_{\mathrm{k}} = \hat{\mathbf{g}}_{\mathrm{LVLH|k}} \otimes \mathbf{g}_{\mathrm{S/C}} \tag{3.145}$$

$$\mathbf{g}_{\mathrm{k+1}} = \hat{\mathbf{g}}_{\mathrm{LVLH|k+1}} \otimes \mathbf{g}_{\mathrm{S/C}} \tag{3.146}$$

The transformation is done by using CRP multiplication which is defined as

$$\bar{\mathbf{g}} = \left(\mathbf{g}_1 + \mathbf{g}_2 - \mathbf{g}_1 \times \mathbf{g}_2\right) / \left(1 - \mathbf{g}_1 \bullet \mathbf{g}_2\right) \tag{3.147}$$

where $\mathbf{g}_1$ and $\mathbf{g}_2$ are two CRP vectors and their resulting product is $\bar{\mathbf{g}}$. Finally, the measurement estimate for the rotation between frames is

$$\hat{\mathbf{g}}_{\mathrm{k|k+1}} = \hat{\mathbf{g}}_{\mathrm{k+1}}^{-1} \otimes \hat{\mathbf{g}}_{\mathrm{k}} \tag{3.148}$$

Thus the measurement estimates in Equation (3.143) are defined by Equations (3.144) and (3.148). The corresponding measurement matrix H is the Jacobian of these equations with respect to the state variables.

$$\mathrm{H} = \begin{bmatrix} \dfrac{\delta \mathbf{t}}{\delta \hat{\mathbf{X}}^*} \\ \dfrac{\delta \mathbf{g}}{\delta \hat{\mathbf{X}}^*} \end{bmatrix} \tag{3.149}$$

The measurement covariance for OLAE was defined in Section 3.6.

3.18.3. IMM Consider VSD Filter

Utilizing the previously defined state dynamics and observation model leads to the formulation of the inspector localization algorithm. The architecture of the algorithm is shown in Figure **3-26** for reference in the following discussion.

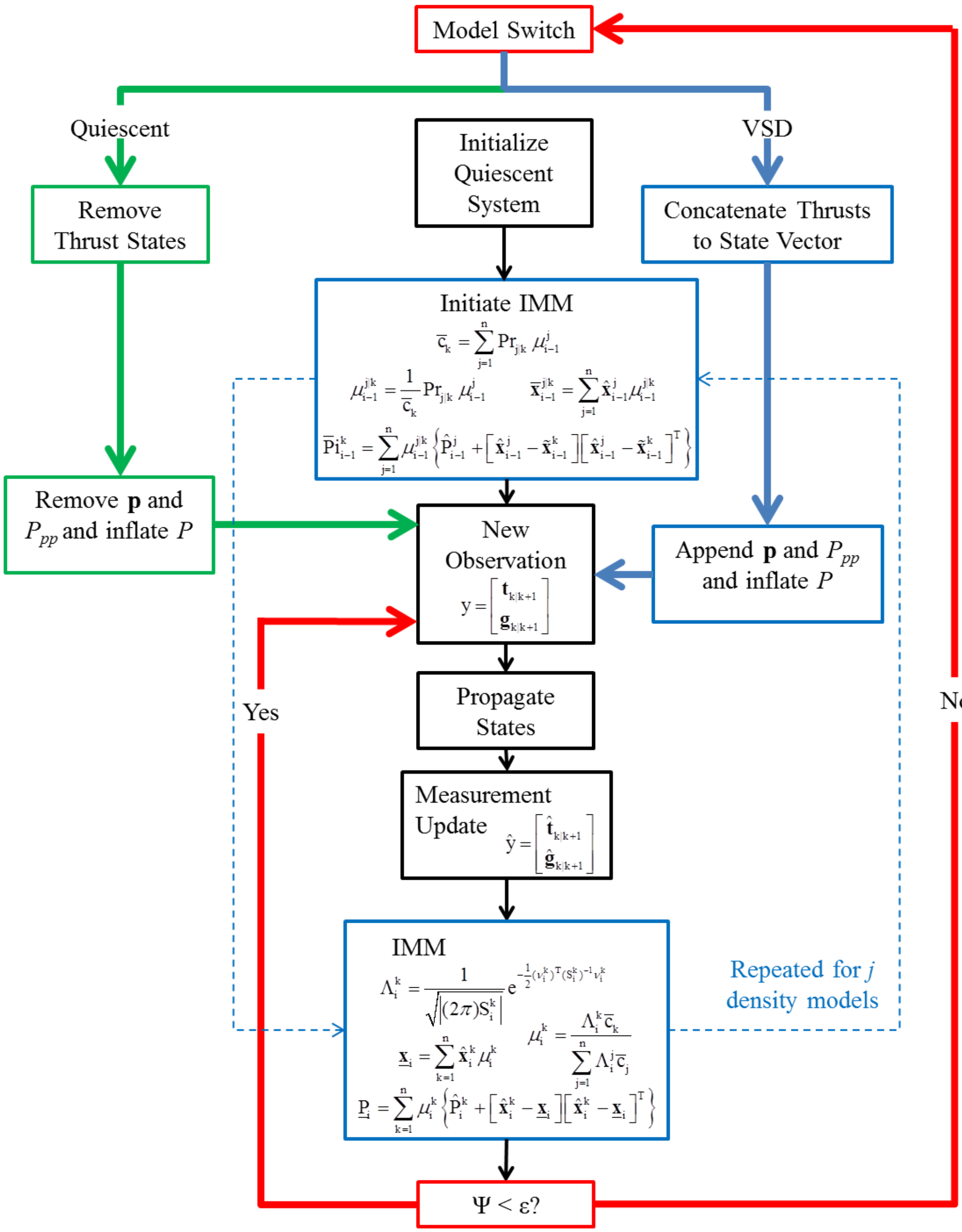

Figure **3-26**: IMM Consider VSD Filter architecture

The estimation process begins once two observations have been made. This is due to the fact that the translation and rotation between frames can only be computed once observations from two frames exist. Once the observations are available, the system is initialized by the available information provided by OLAE. In Figure **3-26**, this would be the block close to the top named initialize quiescent system. The quiescent system follows the flow of the black arrows and black boxes which is a MEKF using the previously defined state estimates and observation model. Refined state estimates are then used to build the octree map and extrapolate the volume estimate and geometric center of the target. This process is iterative and will continue until a maneuver occurs.

Maneuvers made by the target (controlled or uncontrolled) occur at unknown times, duration, thrusting direction and magnitude. At the end of each loop through the filter, the Mahalanobis distance is computed and compared to the pre-defined threshold ε to check if a maneuver occurred. Since the quiescent states do not model thrusting, their associated dynamics are absent thus causing errors to grow in the filter. This is reflected in growing residuals. Depending on the magnitude of thrust, noise present in the system and observations, the lag between maneuver commencement and detection will vary. Once a maneuver is detected, a model switch occurs to include the thrust states. This is depicted in Figure **3-26** by the red arrows. It is at this point that the consider filter and IMM are introduced. At the time the maneuver is detected, the consider parameter **p** is defined as

$$\mathbf{p} = v \tag{3.150}$$

where v is the volume estimate from the octree map. The consider parameter covariance is

$$\mathrm{P}_{\mathrm{pp}} = U \tag{3.151}$$

in which U is the Shannon entropy metric defined in Section 3.7. The consider parameter and its associated covariance are initialized as the current estimate of the volume and Shannon entropy respectively. As per the requirement that these parameters remain unchanged, the initialized values remain constant throughout this procedure. Though if a maneuver ends and a second begins, the consider parameter and covariance may be re-initialized.

Covariance of the state estimate is inflated to account for increased uncertainty in the state estimates as a result of using the quiescent model for an unknown duration to estimate thrust dynamics. Following the blue arrows in Figure **3-26** steps through the IMM consider VSD process. After the thrust states are concatenated to the state vector and the consider parameter and covariance are initialized, the filter initializes the IMM. The procedure begins as outlined in Section 3.14 in which the mixing weights (equally weighted at time t_0) are computed as well as the bank of state estimates and covariances equal to the number of models j. In this dissertation, the number of models is equal to the number of density models in Table **3-3**. After the IMM is initialized, the filter proceeds as normal through the black colored portions of Figure **3-26**. This occurs j times for each density. Since a consider parameter is present after the model switch, the state dynamics, state estimates, and covariance are affected as shown in Figure **3-11**. The associated uncertainty in the consider parameter is reflected in updating the state estimate and covariance. The consider parameter STM is defined as the Jacobian of the nonlinear state dynamics with respect to the consider parameter.

$$\Theta = \begin{bmatrix} \mathbf{0}_{3\text{x}1} \\ \dfrac{-\mathrm{T}_\mathrm{x}}{\rho \mathrm{v}^2} \\ \dfrac{-\mathrm{T}_\mathrm{y}}{\rho \mathrm{v}^2} \\ \dfrac{-\mathrm{T}_\mathrm{z}}{\rho \mathrm{v}^2} \\ \mathbf{0}_{12\text{x}1} \end{bmatrix} \tag{3.152}$$

The variables T_x, T_y, and T_z are the thrust components in each direction in the LVLH frame. The density and volume are denoted by ρ and v respectively. Each j^{th} time through the filter, a different density is used thus changing the thrusting acceleration on the dynamics and directly affecting the resulting state estimate and covariance.

After the filter has run j times, the mixing weights are recomputed based on the residual and residual covariance. The state estimates and associated covariance for each j model are weighted and combined to form the complete state and covariance estimate at the current time step. The Mahalanobis distance is also weighted and combined in the same manner. Since the mixing probabilities are updated and used in the next step, this allows the filter to converge to the density model that provides the best agreement to the dynamics. By providing a coarse volume estimate from the octree map, the filter utilizes the consider parameter and covariance within an IMM architecture to both consider the volume uncertainty and provide the filter with enough density information to converge. Since the density of the target is unobservable and unknown, including a bank of densities and making the evenly distributed density assumption is necessary for filter performance. Since any single density guess could cause a biased or divergent filter, including multiple options allows the filter to choose which combination of densities work best. By indirectly extrapolating mass information from coarse volume estimate and a bank of densities, this allows the target mass to be utilized without directly estimating it in an online filter process. Since mass is also unobservable, estimating it directly would cause divergence. By approaching it indirectly, this is avoided and provides enough information for the estimation filter to converge.

After a given maneuver ends, the thrust model dynamics begin to cause a discrepancy in the estimates. Residuals begin to grow and once the threshold is reached, the model switches again to the green colored portion of Figure **3-26**. The consider parameters are removed as well as the thrust states. Thrust state covariance terms are removed from the state covariance and the remaining covariance is inflated to account for the uncertainty in estimating non-thrusting dynamics with a thrust model. Quiescent model resumes and the IMM is no longer utilized unless another maneuver occurs and is detected.

3.18.4. Guidance and Control

After the current state estimates are updated, guidance and control law is implemented to direct the inspector's attitude to center the target in the left camera frame. The control law is based on the Lyapunov quaternion error control in Sidi[98] and Markley[55]. The quaternion error between the inspector spacecraft body frame *S* and the target body frame *T* is defined as

$$\mathbf{q}_{\mathrm{E}} = \mathbf{q}_{\mathrm{S}}^{-1}\mathbf{q}_{\mathrm{T}} = \begin{bmatrix} \mathrm{q_{T4}} & \mathrm{q_{T3}} & -\mathrm{q_{T2}} & \mathrm{q_{T1}} \\ -\mathrm{q_{T3}} & \mathrm{q_{T4}} & \mathrm{q_{T1}} & \mathrm{q_{T2}} \\ \mathrm{q_{T2}} & -\mathrm{q_{T1}} & \mathrm{q_{T4}} & \mathrm{q_{T3}} \\ -\mathrm{q_{T1}} & -\mathrm{q_{T2}} & -\mathrm{q_{T3}} & \mathrm{q_{T4}} \end{bmatrix} \begin{bmatrix} -\mathrm{q_{S1}} \\ -\mathrm{q_{S2}} \\ -\mathrm{q_{S3}} \\ \mathrm{q_{S4}} \end{bmatrix} \tag{3.153}$$

where $\mathbf{q}_{\mathrm{E}}$ is the quaternion error and subscripts S and T denote the quaternion in the inspector and target frames respectively. Subscripts 1, 2, 3, 4 denote the respective quaternion component in $\mathbf{q}_{1:3}$ and q_4. It is desired to achieve regulation control in which the attitude is brought to a fixed location with zero angular velocity. Starting with the quaternion kinematics (Equation 3.90) and rigid body dynamics defined in Equation (3.3) and the error quaternion in Equation (3.123), the derivative of Equation (3.123) and substituting it into Equations (3.124) and (3.125) leads to

$$\delta\dot{\mathbf{q}}_{1:3} = \frac{1}{2}\left[\delta\mathbf{q}_{1:3}\times\right]\omega + \frac{1}{2}\delta \mathrm{q}_4\omega \tag{3.154}$$

$$\delta\dot{\mathrm{q}}_4 = -\frac{1}{2}\delta\mathbf{q}_{1:3}^{\mathrm{T}}\omega \tag{3.155}$$

A simple feedback controller can simply be defined as

$$\mathrm{L} = -k_{\mathrm{p}}\delta\mathbf{q}_{1:3} - k_{\mathrm{d}}\omega \tag{3.156}$$

where k_p and k_d are positive scalar gains. Though it meets the stability criteria using Lyapunov's direct method, there is no guarantee that the shortest path is followed to reach the final desired attitude. A better non-linear control law which achieves the shortest distance is

$$\mathrm{L} = -k_p \mathrm{sign}(\delta \mathrm{q}_4)\delta \mathbf{q}_{1:3} - k_d (1 \pm \delta \mathbf{q}_{1:3}^{\mathrm{T}} \delta \mathbf{q}_{1:3})\omega \tag{3.157}$$

This is the control law used in the inspector localization algorithm. Since developing control law is not a focus of this dissertation, using a straightforward proven approach was desired.

The guidance of the inspector's control is determined by the attitude of the target spacecraft, geometric center of the target, and the estimated offset between geometric center and center of mass. The target attitude and geometric center is extrapolated from the octree map and is utilized to determine the guidance (desired quaternion) for the control law.

3.18.5. Inspector Localization Algorithm

Compiling the procedures and algorithms from Sections 3.18.1 through 3.18.4 defines the full inspector localization algorithm. Figure **3-27** shows the flow of the approach beginning with the image generation. After an image in the current time step is generated, the image is sent to the image processing algorithm (KLT) to extract a sparse set of features in pixel coordinates.

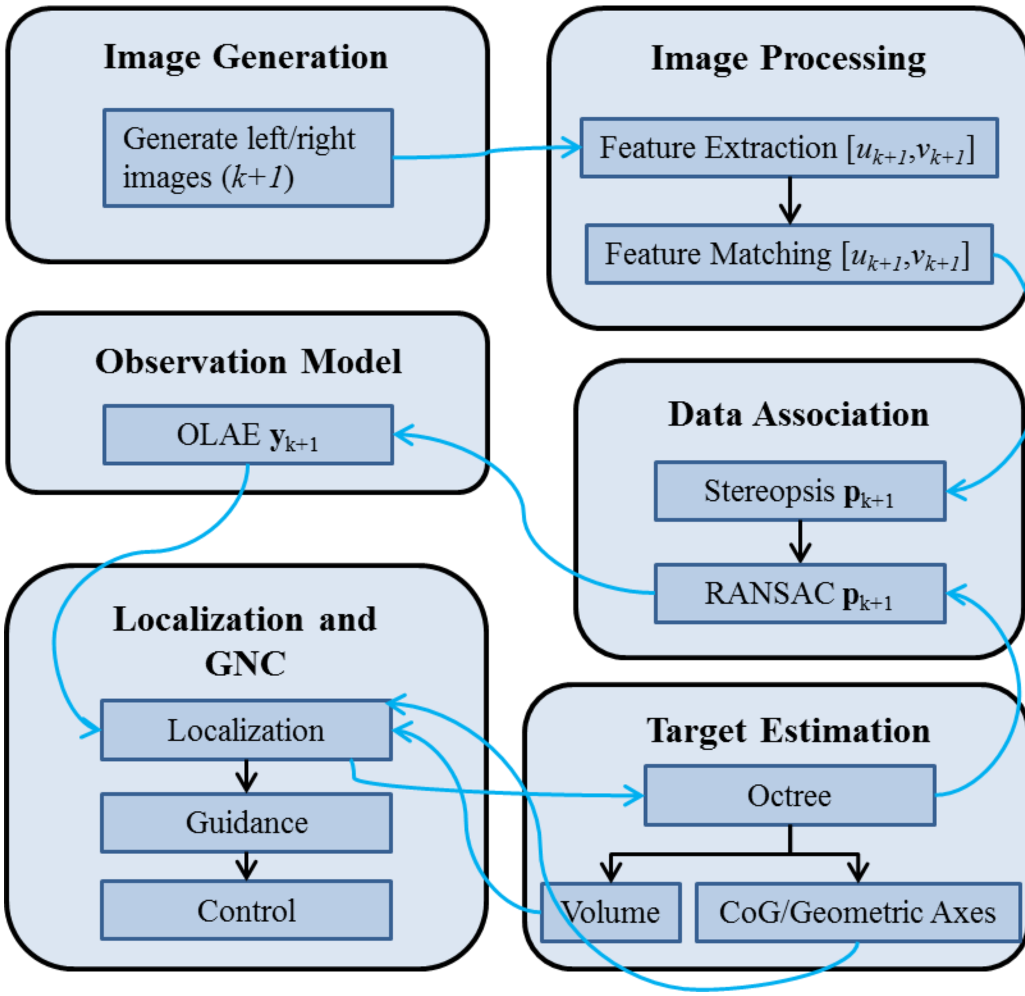

Figure **3-27**: Inspector localization algorithm

Between the left and right images, matches are found and triangulated in the stereopsis process. RANSAC is performed to eliminate the outliers and the remaining inliers are used to compute the rotation and translation between the current and previous point clouds using OLAE. This forms the basis of the observation model and thus the inspector localization algorithm is initialized to smooth out these parameters. An octree map is formed from the state estimates and feature points to extrapolate volume, geometric center, and geometric axes. The octree map itself is used to improve the RANSAC performance after the confidence in the map is high enough. Volume and the geometric center are fed into the localization algorithm to provide the foundation for maneuver detection and estimation. Finally, the geometric axes and center of geometry from the octree are used in the guidance and control algorithm to aim the camera's boresight toward the target. Iteration continues until no new images are taken. This framework will be tested in simulated experiments in Chapter 4.

Chapter 4

Results and Analysis

4.1. Simulation Setup

The following sections will define the setup of the simulated scenarios. These will cover the assumptions and initialization of the scenario orbital regime, stereo camera optical parameters, and information about the target spacecraft. Two scenarios are analyzed in which the thrust magnitude is different by a factor of ten. This is done to determine how the algorithm handles different magnitudes of thrust in regards to detection and estimation. For each scenario, two cases are conducted. The first case involves initiating the maneuver after the observer has developed a robust, low uncertainty octree map. The second case initiates the maneuver before the target has been fully mapped causing the octree mapping to be sub-optimal. These cases are compared to analyze the effects of the octree mapping on the filter performance.

4.1.1. Simulation Parameters

The deputy spacecraft is placed in a natural motion circumnavigation (NMC) orbit around the target (chief) satellite. This type of orbit shown in Figure **4-1** is conducive for conducting an inspection. Initially, the target is located at the center of the NMC and lies at the origin of the LVLH frame. The initial conditions for the deputy and chief satellites are in Table **4-1**. The LVLH and ECI frames are initially aligned at the start of the simulation. As time propagates, the LVLH frame rotates about the common **z** axis at an angular rate equal to the mean motion of the orbit.

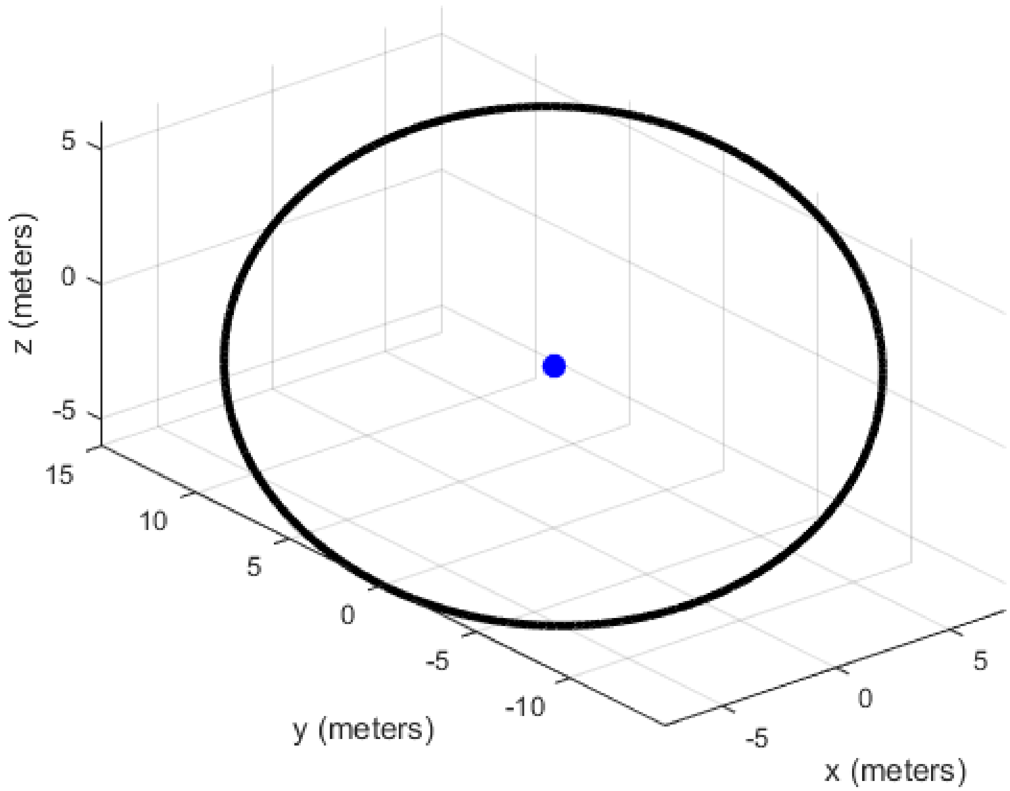

Figure **4-1**: Natural motion circumnavigation orbit in LVLH Frame

After *500* seconds into the simulation, a continuous thrusting maneuver occurs in the *y* direction of the LVLH frame for *500* seconds; the simulation continues for an additional *150* seconds after the thrusting maneuver ends. A thrusting duration of *500* seconds was chosen so the estimation filter has an opportunity to detect and estimate the maneuver. Additional time is added after the maneuver to analyze when the maneuver end detection occurs and how the filter copes post-maneuver. Lighting conditions are more optimal at the start of the simulations and become less so as it progresses. This was set up to have varying lighting conditions for the filter to handle. At simulation start, the sun vector is illuminating the object; however, towards the end, the trajectories begin to enter the shadow of the Earth. If the simulation were to continue beyond its timeframe, the sun would not have line of sight with both spacecraft.

Table **4-1**: Orbital initial conditions for the Deputy (a) and Chief (b)

(a)

	Deputy (LVLH)
Position (meters)	[-7.5, 0, -6]
Velocity (m/s)	[0, 0.0165, 0]

(b)

	Chief (ECI)
Position (meters)	[6906385, 0, 0]
Velocity (m/s)	[0, 7597, 0]

Two scenarios are conducted in which the magnitude of the thrusting maneuver changes. In each scenario, two cases are run in which the observing satellite has already completed an initial orbit and thus has developed a converged mapping of the target before the maneuver begins. The second case is that the maneuver occurs before the observing satellite has fully developed a complete mapping of the object. As described in Chapter 3, the estimate volume and associated covariance are initialized as the consider parameters upon the detection of a maneuver and remain unchanged for the duration of the detected maneuver. The purpose of running these two cases are to determine the effectiveness on filter performance of a converged mapping versus a sub-optimal mapping. Summarized in Table **4-2** are the simulation parameters utilized in these simulations.

Table **4-2**: Simulation Parameters

	Scenario 1	**Scenario 2**
Simulation Duration	1150 seconds	1150 seconds
Thrust Start Time	500 seconds	500 seconds
Thrust End Time	1000 seconds	1000 seconds
Thrust Magnitude	0.465 mm/s^2	0.0465 mm/s^2

4.1.2. Stereo Camera Initialization

Observations of the deputy spacecraft are conducted via two stereo cameras with identical intrinsic properties (Table **4-3**). Camera parameters were arbitrarily chosen. The baseline of the cameras is the center to center distance between both cameras. Rotation describing the spacecraft to camera frame is a known quantity and assumed to be constant. The pixel spacing defines the center to center distance between pixels in the generated image. Random pixels encompassing the target had its brightness randomly adjusted to add noise to the images.

Table **4-3**: Camera properties[‡]

Resolution [width, height] (pixels)	[2400, 1900]
Optical Center [width, height] (pixels)	[1200, 850]
Focal Length	20 *mm*
Camera Screen Size	28.2 *mm* (along diagonal)
Baseline	0.6 *m*
Spacecraft to Camera Frame	$\begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
Pixel Spacing	9.2125 μm
Horizontal Field of View	47.268 °

4.1.3. Target Model Parameters

The target spacecraft was created using SOLIDWORKS and is shown in Figure **4-2**. Model properties including those from Table **4-4** were obtained from the SOLIDWORKS Mass Properties command manager. These properties are computed in SOLIDWORKS based on how the model is constructed.

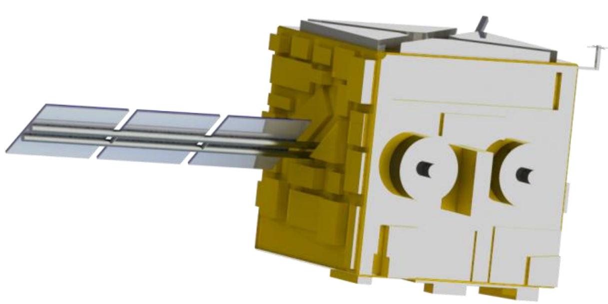

Figure **4-2**: Spacecraft model

[‡] Camera properties are not representative of any particular camera

It should be noted that the SOLIDWORKS model is solid so the mass, center of mass, and moment of inertias are computed in SOLIDWORKS as such. All parameters are in the target body Frame. Center of Mass is positioned at [0, 0, 0] *meters* in the LVLH Frame. Since spacecraft are not solid objects, the values in Table **4-4** are assumed to be the best approximation for a satellite of this shape.

Table **4-4**: SOLIDWORKS model parameters

Mass of target	215 kg
Center of Mass	[0.005, -0.0131, 0.0127] *m*
Principal Moment of Inertias	[469.141, 503.221, 520.301] $kg\ m^2$

The spacecraft model was created to be generic containing a solar array, two optical cameras on the front, and an antenna on the upper right. Each surface of the spacecraft body was given further detail for the simulated cameras to detect.

4.1.4. Gyroscope and Bias Initialization

The gyro model is taken from Markley[55] and is written as

$$\boldsymbol{\omega} = (I_3 + S^{True})\boldsymbol{\omega}^{True} + \mathbf{b}^{True} + \eta \tag{4.1}$$

$$S^{True} = \begin{bmatrix} s_1^{True} & k_{U1}^{True} & k_{U2}^{True} \\ k_{L1}^{True} & s_2^{True} & k_{U3}^{True} \\ k_{L2}^{True} & k_{L3}^{True} & s_3^{True} \end{bmatrix} \tag{4.2}$$

where I_3 is a 3x3 identity matrix, $\boldsymbol{\omega}$ and $\boldsymbol{\omega}^{\text{True}}$ are the measured and true angular velocities, respectively. Scale factors are denoted by $\mathbf{s}^{\text{True}}$ with vector components s_1^{True}, s_2^{True}, and s_3^{True}. The true bias is $\mathbf{b}^{\text{True}}$ and gyro misalignments are given by $k_{\text{U}}^{\text{True}}$ and $k_{\text{L}}^{\text{True}}$. The spectral density is denoted as η. Noise added to the gyro measurements is increased when the attitude changes rapidly.

The bias itself is modeled as a discrete-time linear system. It can be written in the form

$$x = Ax + Bu \tag{4.3}$$

$$y = Cx + Du \tag{4.4}$$

where u is an input to the system and matrices A, B, C, and D are the state space representation for a single-input transfer function. The transfer function is given by

$$H(s) = \frac{B(s)}{A(s)} = \frac{b_1 s^{n-1} + \cdots b_{n-1} s + b_n}{a_1 s^{m-1} + \cdots a_{m-1} s + a_m} = C(sI - A)^{-1}B + D \tag{4.5}$$

where a is a vector containing coefficients in descending powers of s. Rows of matrix contain vectors of numerator coefficients in descending powers of s. Initial conditions for the gyro bias were taken from Markley[55] and are summarized in Table **4-5**. It should be noted that ppm denotes parts per million.

Table **4-5**: Gyroscope initial parameters[55]

Initial Gyro Bias	*0.1* deg/h for each axis
Gyro Scale Factors	s_1 = 1500 ppm, s_2 = 1000 ppm, s_3 = 1500 ppm
Gyro Misalignments	k_{U1} = 1000 ppm, k_{U2} = 1500 ppm, k_{U3} = 2000 ppm k_{L1} = 500 ppm, k_{L2} = 1000 ppm, k_{L3} = 1500 ppm
Coefficients	a = [2, -2] b = [1, 1]

4.1.5. Scenario Initialization

Utilizing the aforementioned parameters defining the simulation including the trajectories, camera properties, and the target model, the system is initialized for scenarios 1 and 2 according to Figures **4-3** and **4-4** respectively. In both of these figures, the deputy is visualized in black and the target spacecraft in blue. Lines emanating from the spacecraft models represent the trajectory over the simulation duration in the LVLH frame. Attached to the deputy spacecraft is a semi-transparent prism cone representing the field of view of the stereo cameras. The prism cone is merely a visual representation so the range extends beyond the edge of the prism encompassing the target. The dashed blue, green, and red lines define the camera axes for both cameras.

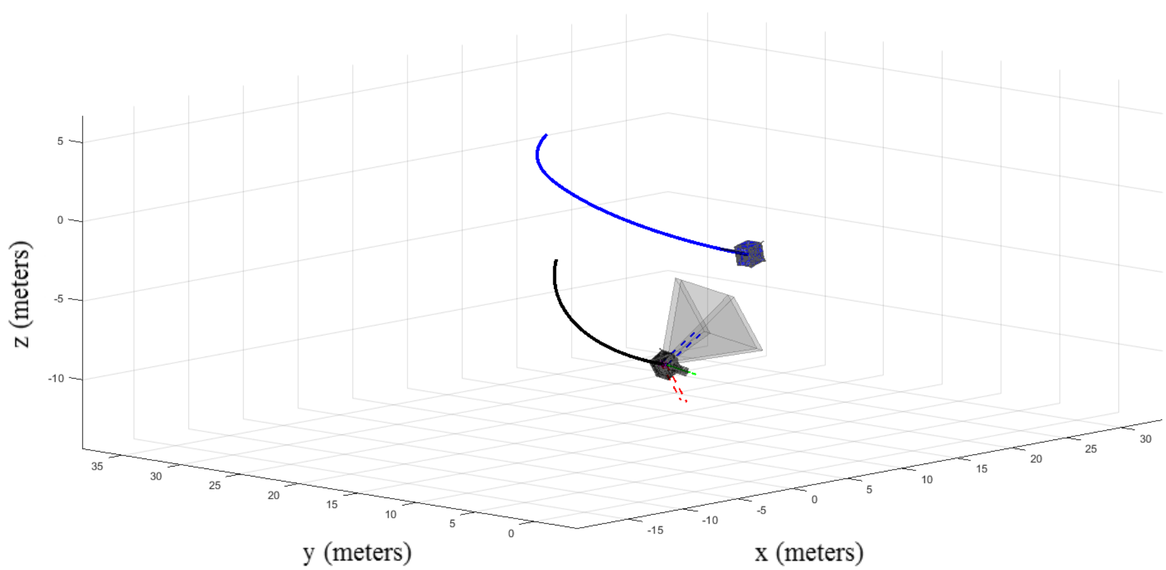

Figure **4-3**: Simulation start of Scenario 1

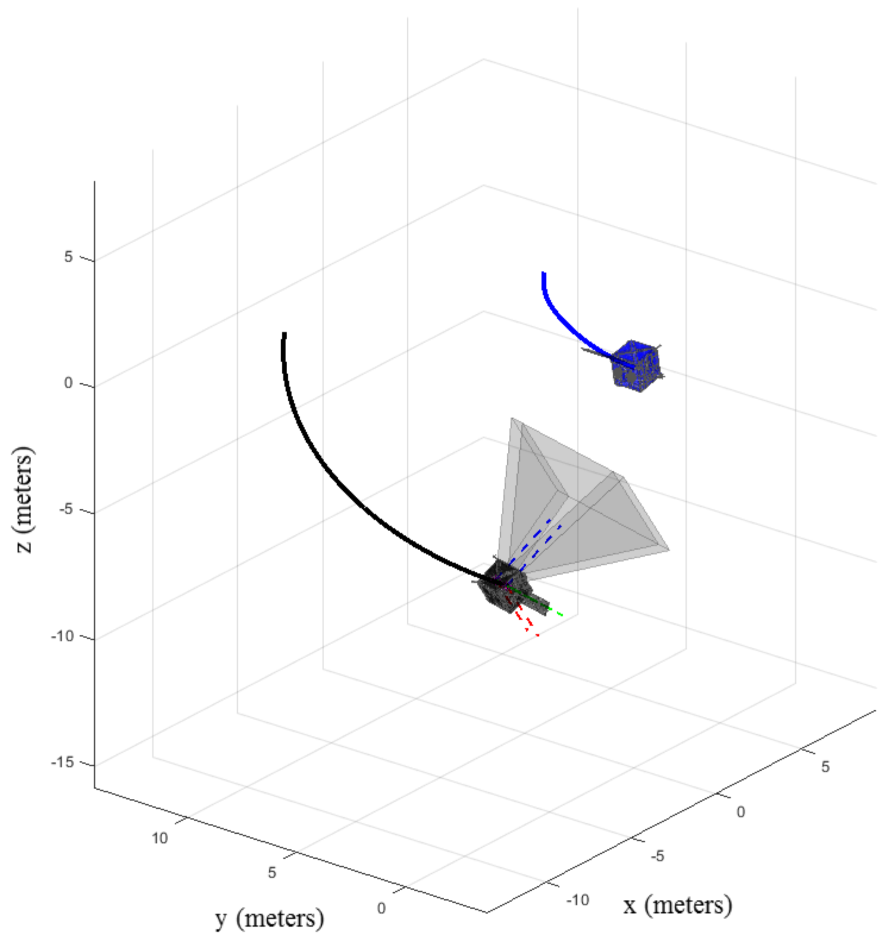

Figure **4-4**: Simulation start of Scenario 2

Initial conditions are randomized for the inspector spacecraft (Deputy) at the start of the simulation. The positional states in Table **4-1(a)** had random Gaussian noise with a standard deviation of *3.5 meters* applied to those initial conditions. The velocity states in Table **4-1(b)** added random Gaussian noise with a standard deviation of *0.05 meters/second* to initialize the velocity states for the estimation filter. The true attitude quaternion for the inspector in the LVLH to inspector spacecraft body frame is [*0*, *-0.3*, *-0.1*, *1*]. Gaussian noise with a standard deviation of *0.2* is added to each element of the quaternion and normalized. Both the geometric to center of mass offset vector and thrust acceleration states are initialized as a *3*x*1* vector of zeros. Gyro bias is initialized as described in Section 4.1.4. The covariance is initialized as

$$P_{r_{LVLH}} = \begin{bmatrix} 400 & 0 & 0 \\ 0 & 400 & 0 \\ 0 & 0 & 400 \end{bmatrix} \quad P_{v_{LVLH}} = \begin{bmatrix} 60 & 0 & 0 \\ 0 & 60 & 0 \\ 0 & 0 & 60 \end{bmatrix} \quad P_{g_{LVLH/S}} = \begin{bmatrix} 40 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 40 \end{bmatrix}$$

$$P_b = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad P_{a_{th}} = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} \tag{4.6}$$

where each *3*x*3* matrix corresponding to the states in the state vector from Equations (3.140) and (3.141) form a *12*x*12* for the quiescent states and *15*x*15* for the thrust model, respectively.

4.2. Feature Matching and Three-Dimensional Point Cloud

At any given observation time step, two images are generated for each camera. A sparse feature set is extrapolated from each image and matched according to the procedure in Section 3.3. In Figure **4-5**, an image from the left and right cameras are shown with the full extracted features (depicted by red dots) and the corresponding matched features (represented by yellow circles). The pixel coordinates of these stereo correspondences are triangulated to produce three-dimensional features from the camera frame. The set of these features form the three-Dimensional point cloud. However, outliers need to be removed from this set as mismatched feature correspondences will occur. Mismatched features are highly more likely in repeating textures (solar panels) and similar looking structures or components. Any mismatched features that make it through the RANSAC will create bad observations and adversely affect the octree map.

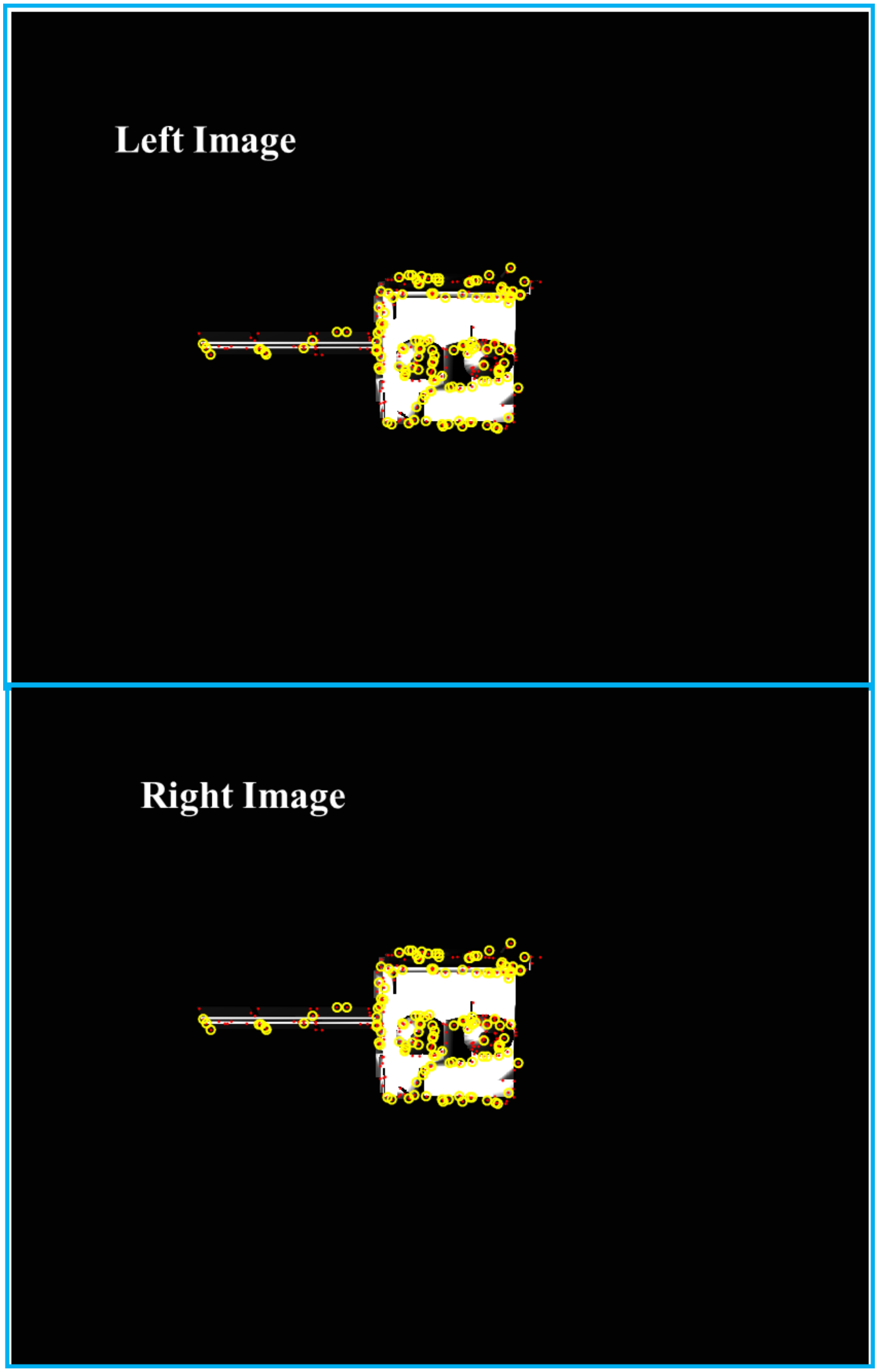

Figure **4-5**: Stereo correspondences between left and right images

Figure **4-6** shows a sample time step in which mismatched features makes it through the RANSAC. In these situations, it is up to the estimation filter to handle periodic bad observations. In the formation of the octree, outliers are further pruned by comparing the dataset to itself and the confines of the bins. After the estimation for the newest observations are complete, the octree map for the current time step is updated into the global octree map. Features overwrite existing matched features if the produced covariance from the estimation filter is better than the previous time step. This ensures that features with better certainty are updated without having to include them directly in the estimation filter. New features that do not currently exist in the octree map are added.

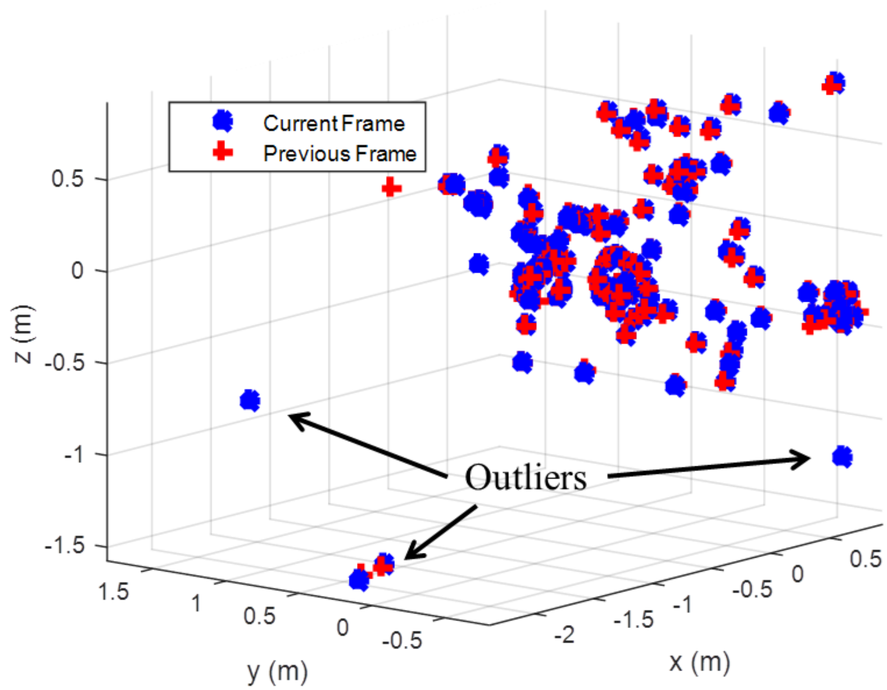

Figure **4-6**: Three-dimensional point cloud with outliers

4.3. Simulated Results

Continuous thrusting maneuvers can be very difficult to detect since their corresponding thrust magnitudes are smaller than impulsive thrusts. Detecting the start and end of a maneuver occurs with a certain amount of lag depending on thrusting magnitude, level of noise present in the measurements, and the tuning parameters of the estimation filter. In a visual based measurement system, the quality of the observations is based upon the lighting in the image and mismatched feature correspondences. Shadows, over-saturation, occlusions, and reflections are dependent on the position of the spacecraft relative to the sun vector. In the image generation used in this dissertation, only the first three effects are modeled. Due to the harsh lighting conditions in the space environment, detected features can be easily lost from one frame to the next. Tuning the maneuver detection properly is vital as making them too sensitive will cause bad measurements to falsely detect a maneuver or its end. Conversely, making the tuning parameters too high will cause maneuvers to go undetected.

Results in the following sections will analyze the filter performance of two different thrusting maneuvers with a very low thrust magnitude. For each thrusting maneuver, the start of the thrusting maneuver was planned such that the convergence of the octree map was converged and incomplete. Robustness of the octree map is correlated to the convergence of the volume estimate obtained from the octree map. In following figures, the volume refers to the estimate volume of the target obtained from the octree map. This is demonstrated to study the effects on map convergence for maneuver detection and estimation for a robust and sub-optimal octree target mapping. A Monte Carlo study was conducted in which *100* simulations were run per octree mapping case (robust and sub-optimal) for each scenario. Thus, *200* total simulations were conducted for each scenario. In each Monte Carlo run, the initial conditions and process noise for the estimation filter are randomized. Furthermore, the generated images have random noise applied to some pixels adding noise to the measurements for every Monte Carlo simulation.

4.3.1. Translational Estimation

Relative position and velocity are estimated in the LVLH frame with the target satellite centered at the origin of the frame. After *500* seconds, the target begins a thrusting maneuver in the **y** direction in the LVLH frame after which the target begins to move from the center of the frame. Upon detecting the maneuver, the estimation filter initializes the volume estimate from the octree map and the associated uncertainty as the consider parameter and covariance respectively. For *100* simulations, the observer has had a chance to complete an entire revolution around the target before the simulation starts. This allowed the observer to develop an octree map of the target with low uncertainty. Conversely, another *100* simulations were conducted in which the observer did not have a previous orbit of the target and thus began building the octree map at the simulation start. Since the maneuver occurs before the observer has an opportunity to image the entire target, the octree map is incomplete. As a result, the volume estimate is more inaccurate with higher covariance at the time a maneuver is detected and the consider filter initialized.

In Figure **4-7**, relative trajectories for the more robust and incomplete octree mapping are plotted against the true relative position for scenario 1. Converged volume refers to the volume estimate of the target obtained from the robust octree target mapping; sub-optimal volume refers to the volume estimate of the target obtained from the incomplete octree target mapping. Trajectory data from the full *100* runs are averaged and plotted. It can be easily seen along the trajectory where the maneuver is detected as the less robust octree mapping (Sub-optimal Volume) begins to diverge from the true trajectory.

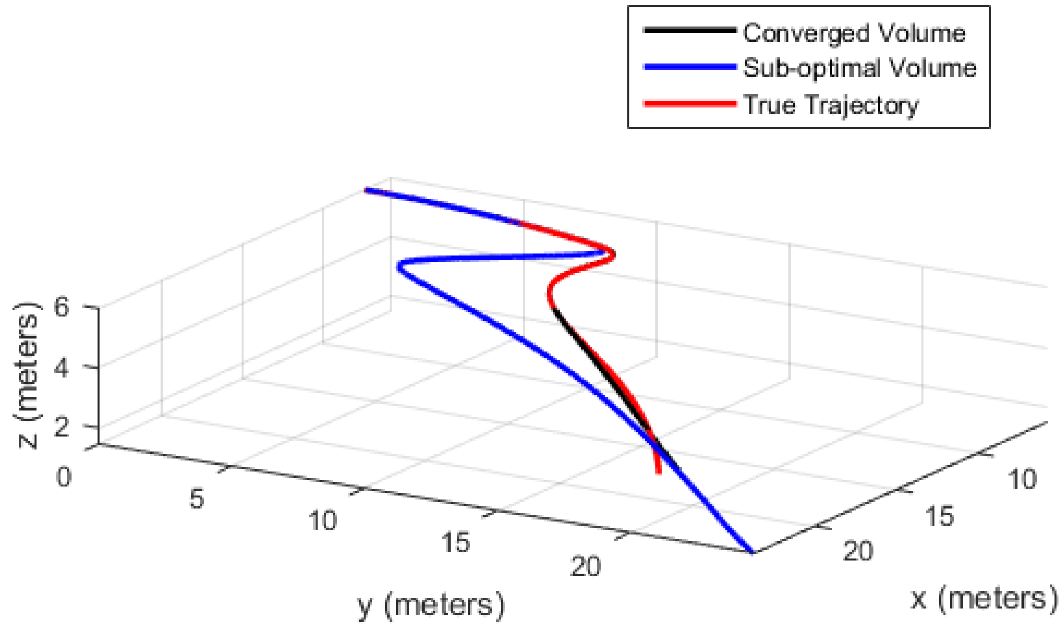

Figure **4-7**: Estimate relative trajectories for 0.465 mm/s^2 thrust in Scenario 1

Both during the maneuver and afterwards, the trajectory estimate does not fully recover. However, the estimate having a better octree mapping remains relatively close to the true trajectory throughout the simulations. In scenario 2 with a lower thrust magnitude, the trajectory estimation is considerably better for the less converged octree target mapping. The estimated trajectories in Figure **4-8** are both very accurate. In this scenario, having a better octree mapping is less important for filter performance. Even with a coarser volume estimate, the simulations conducted with a less robust octree mapping performed similarly to the more converged map.

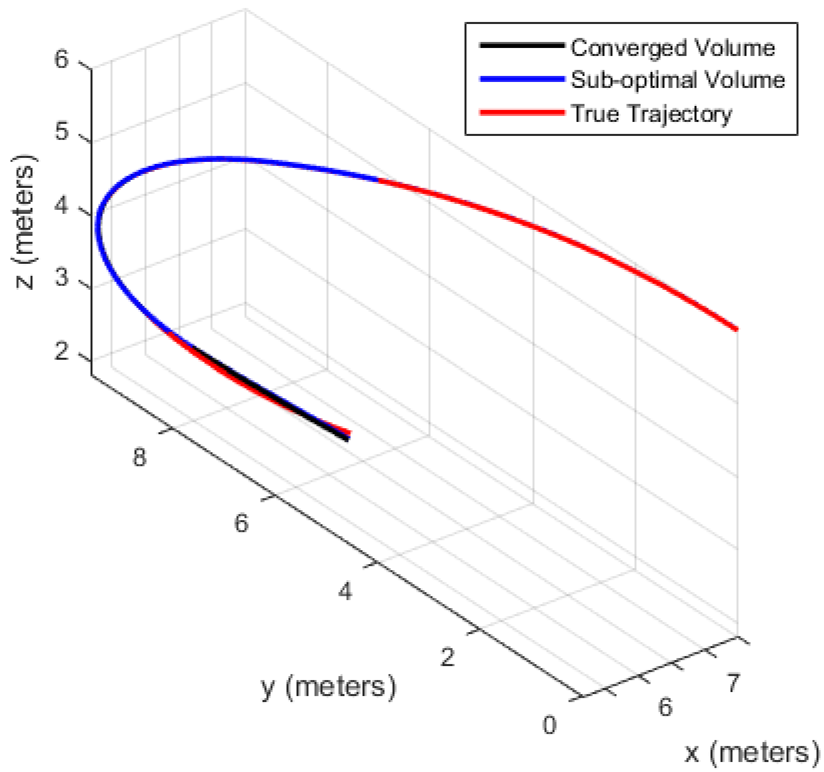

Figure **4-8**: Estimate relative trajectories for 0.0465 mm/s^2 thrust in Scenario 2

Plotting the trajectories versus time in Figures **4-9** and **4-10** better shows the maneuver timing's effects on the estimated trajectories for scenario 1 and 2 respectively. Again, these plots are a result of the averaged dataset for all Monte Carlo runs. In scenario 2, the estimated trajectories have very little deviation from the truth for all positional states. However, the higher thrust magnitude present in scenario 1 shows greater error for the incomplete octree target mapping. After the maneuver initiates at 500 seconds, the estimation begins to grow in error as the thrusting continues. The error continues after the maneuver ends especially in the **x** and **y** positional states.

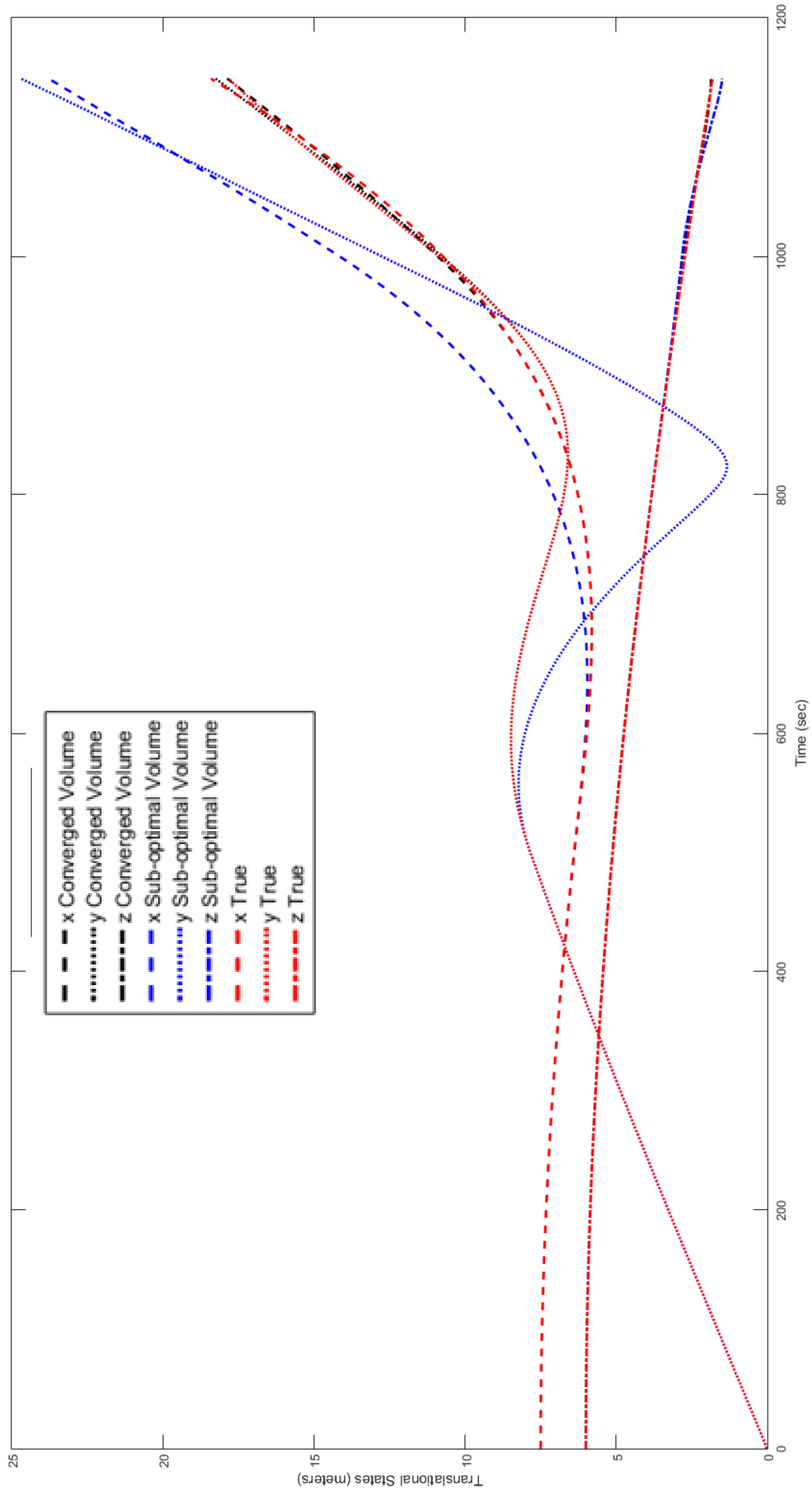

Figure **4-9**: Scenario 1 translational states versus time

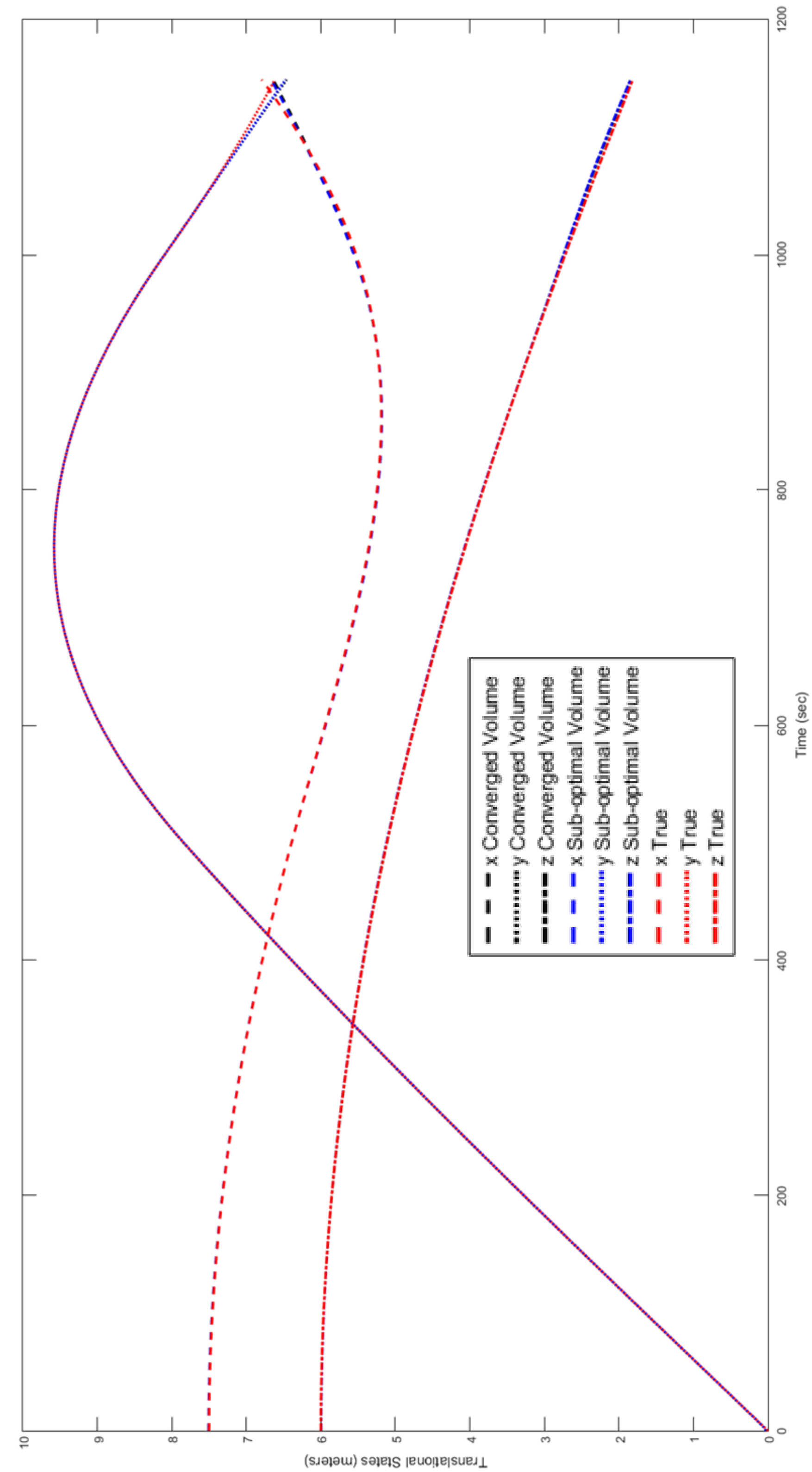

Figure **4-10**: Scenario 2 translation states versus time

Examining the root mean square (RMS) residual positional error for both scenarios provides better explanation for the results. As described in Chapter 3, the transition from the quiescent model to the thrusting model (and vice versa) is accompanied by a covariance inflation up to a pre-defined threshold to account for increased uncertainty in the states from using an incorrect dynamical model to perform the state estimation. The dotted lines in Figures **4-11** through **4-16** represent the 3σ bounds and the moment in which the start and end times of maneuver detection are clearly visible. Around *500* seconds, the covariance bounds have a sharp increase for both scenarios signifying a covariance inflation preceding maneuver detection. Again after *1000* seconds, another increase in the covariance is depicted signifying the detected end of the maneuver.

In scenario 1 (Figures **4-11** to **4-13)**, the difference between the positional errors and associated covariance between both octree mapping cases is noticeable. After the maneuver occurs until the end of the simulation, the positional covariance bounds for the less robust (blue lines) mapping are larger than those for the complete octree mapping (black lines). The RMS residual error of all the positional states is greater for the incomplete octree mapping with the error struggling to remain within the covariance bounds. The covariance of the less robust target mapping has two large spikes in the covariance after *1000* seconds. Since these plots are the RMS of the entire dataset, a large number of simulations detected the end of the maneuver around those time steps; whereas the end maneuver detection for the more robust octree mapping were more varied occurring on average with less lag after the maneuver actually ended.

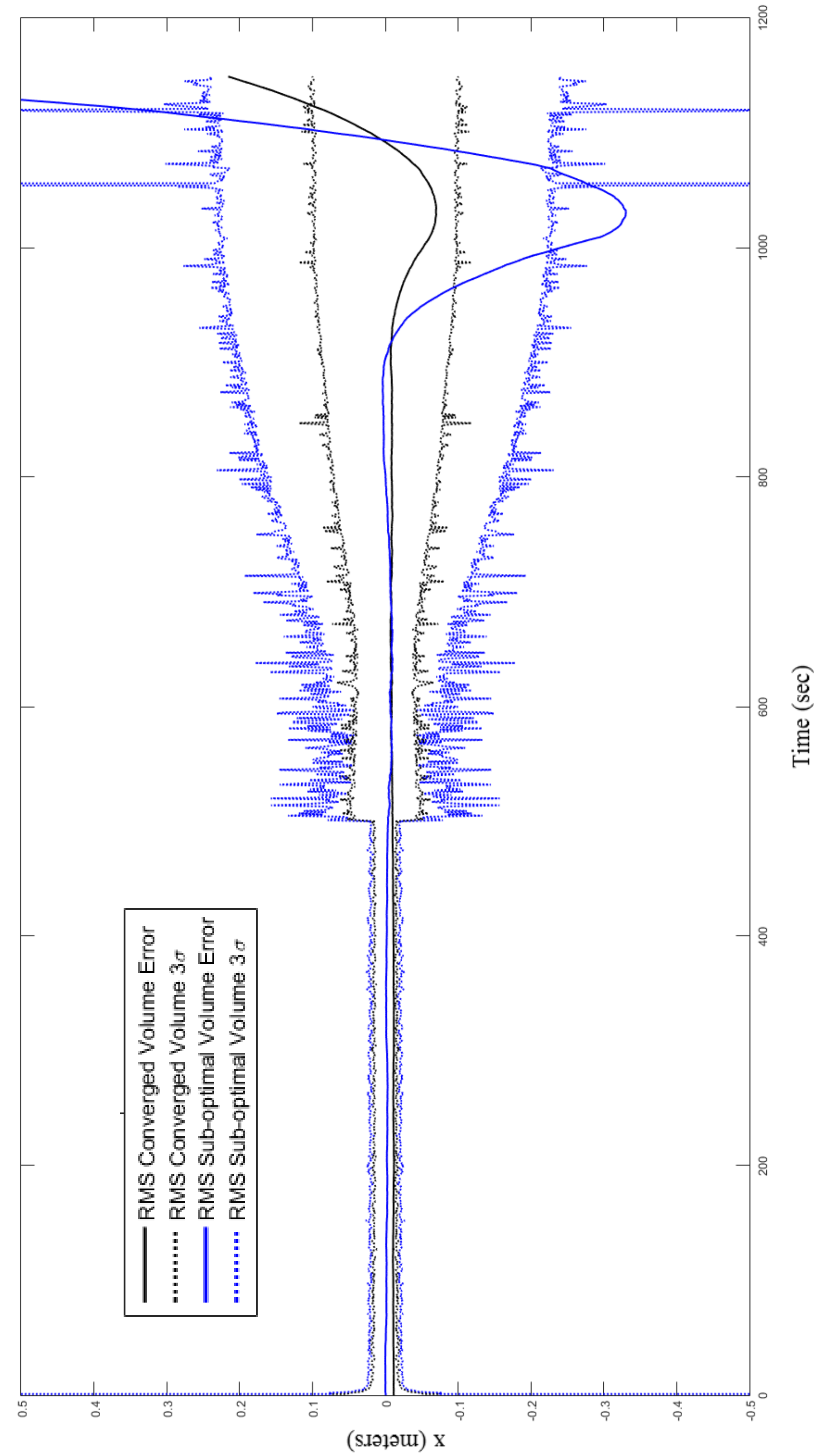

Figure **4-11**: Scenario 1 RMS residual error of the **x** positional state with 3σ covariance bounds

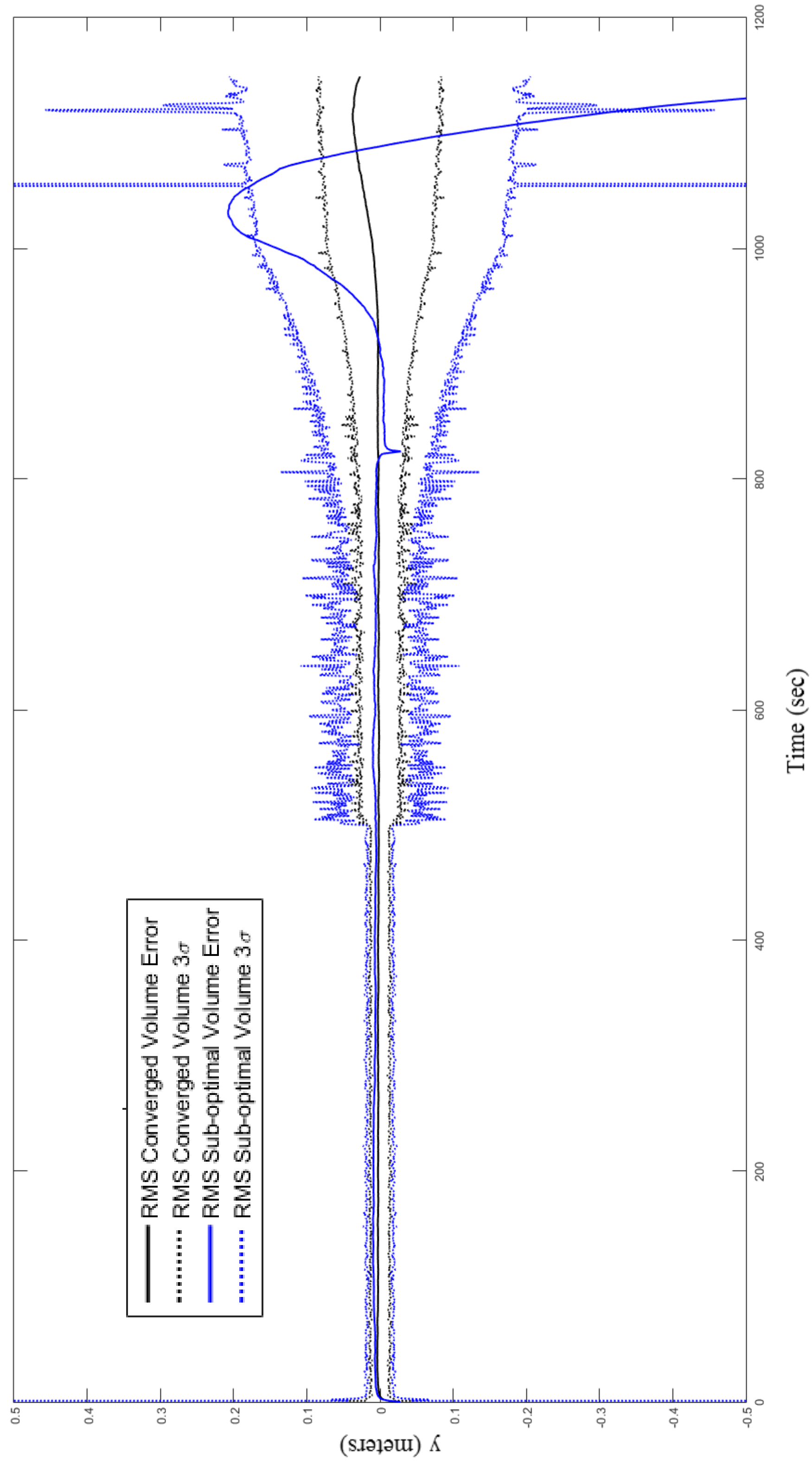

Figure **4-12**: Scenario 1 RMS residual error of the **y** positional state with 3σ covariance bounds

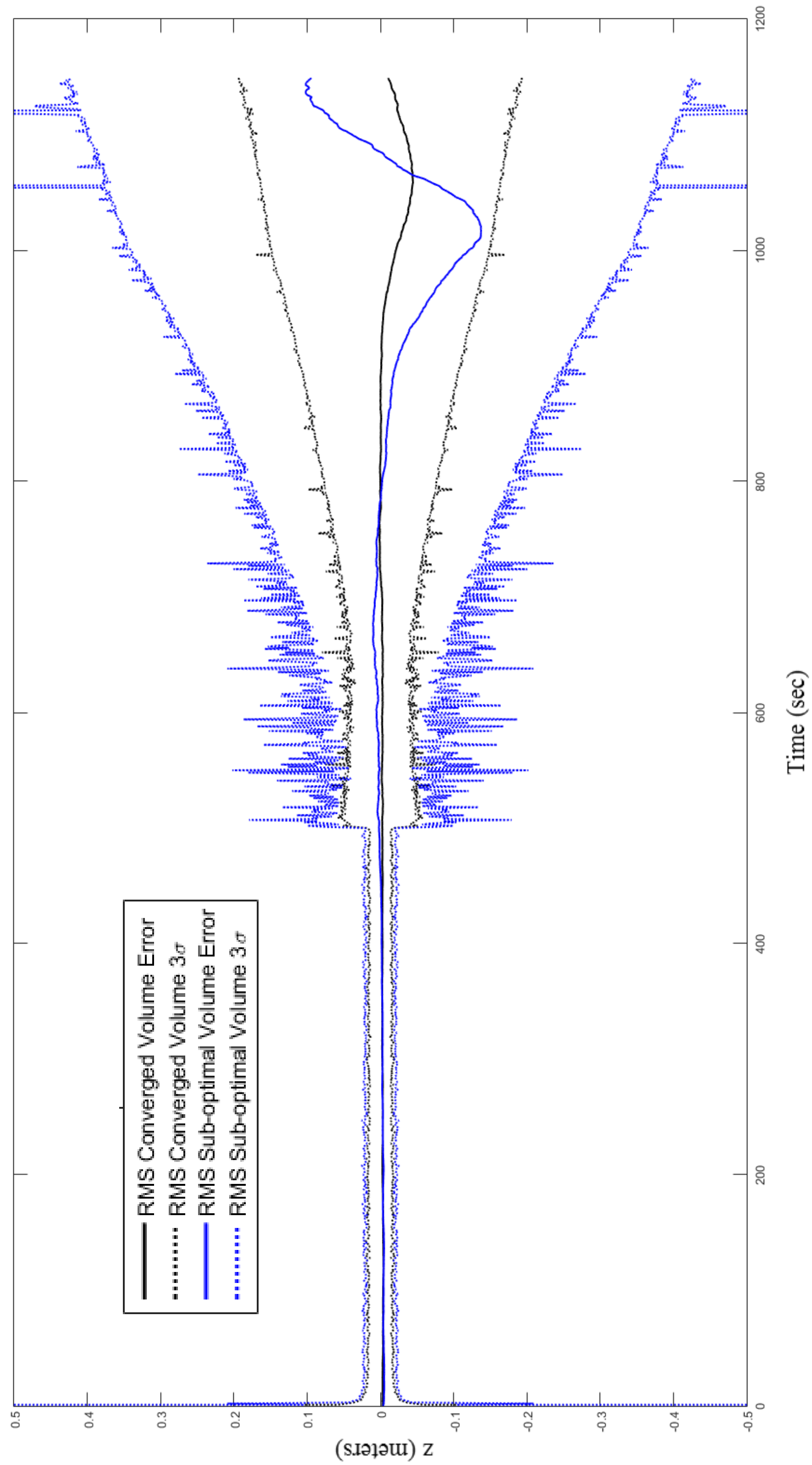

Figure **4-13**: Scenario 1 RMS residual error of the **z** positional state with 3σ covariance bounds

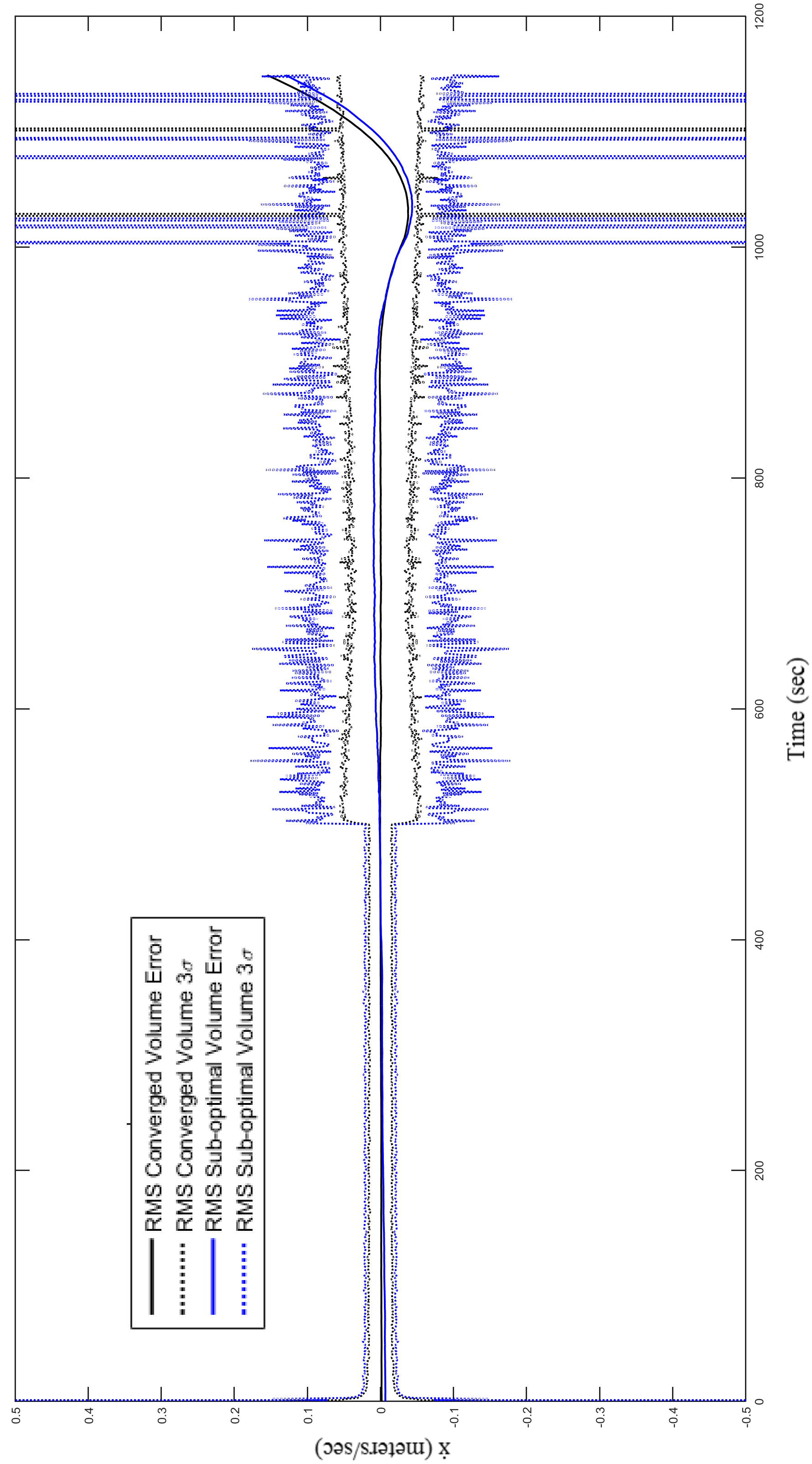

Figure **4-14**: Scenario 2 RMS residual error of the **x** positional state with 3σ covariance bounds

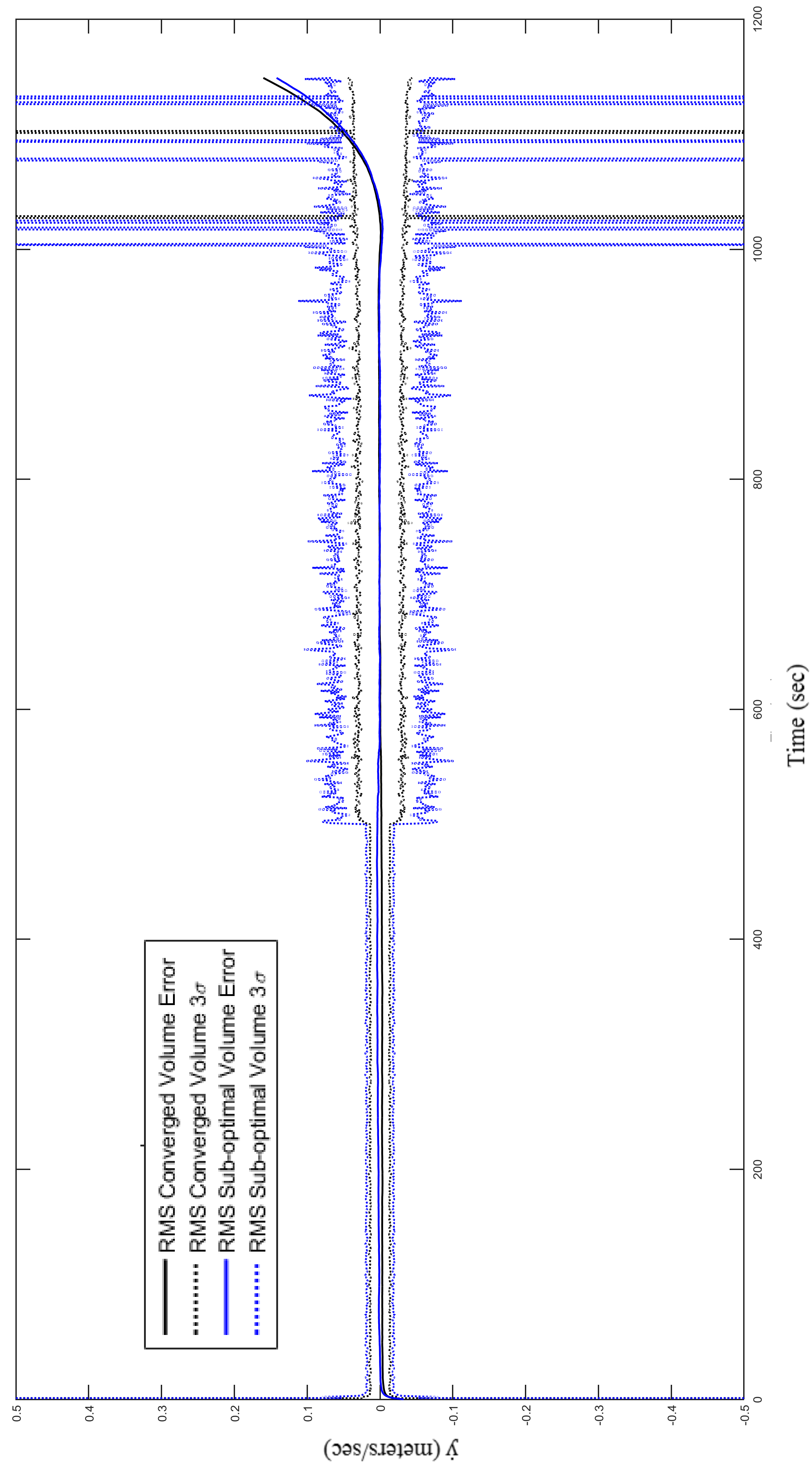

Figure **4-15**: Scenario 2 RMS residual error of the **y** positional state with 3σ covariance bounds

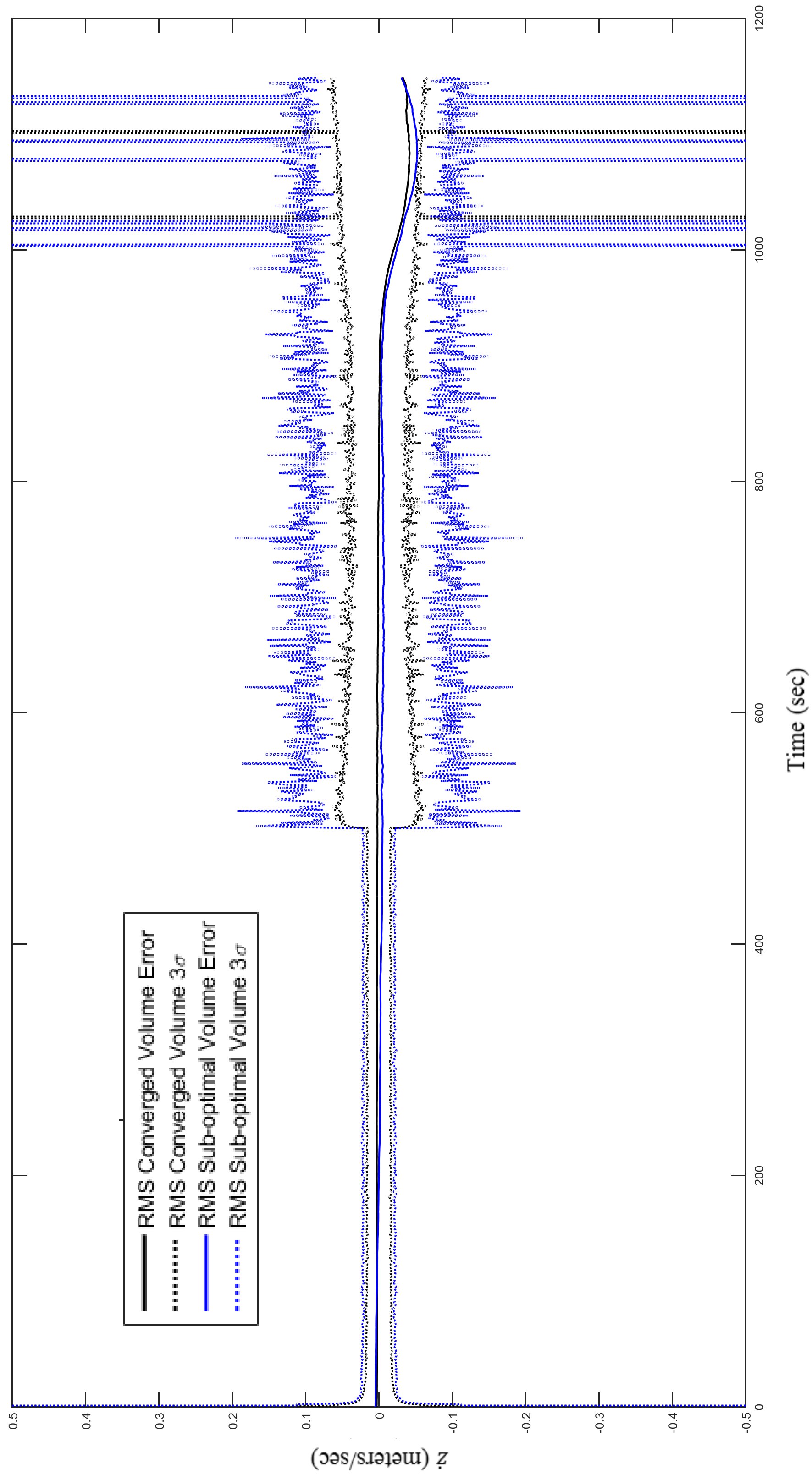

Figure **4-16**: Scenario 2 RMS residual error of the **z** positional state with 3σ covariance bounds

Positional results from scenario 2 have less variance in the RMS residual error. Figures **4-14** through **4-16** show a tighter covariance bound post maneuver for both octree mapping cases. This is expected behavior as the thrust magnitude is an order of magnitude less than that of scenario 1. The covariance for the more robust octree mapping is slightly better than the less robust mapping with similar RMS residual errors. Similar to the less robust target mapping covariance in scenario 1, both octree mapping cases have several large spikes in the covariance after *1000* seconds. These spikes are a result of various maneuver detection times. Since the thrust magnitudes an order of magnitude less than scenario 1, the maneuver start and end are more difficult to detect resulting in a wider variance of detection times.

Relative velocity estimates from both scenarios (Figures **4-17** through **4-22**) larger covariance variations than the positional estimates due to the uncertainty caused in the velocity by the velocity component due to thrusting. In scenario 1 (Figures **4-17** to **4-19**), the velocity estimates for both octree mapping cases remain very stable through most of the simulation. Most of the error occurs during and after the maneuver especially in the less robust octree mapping case which has larger error spikes. Again, around *500* and *1000* seconds, large spikes in the covariance occur with smaller spikes in the covariance surrounding indicating variance in the times the maneuver detection occurred.

Relative velocity estimates in scenario 2 present similar characteristics with the oscillating spikes in the covariance relative to the positional states. Residual RMS errors are smaller than scenario 1 due to the lower thrust magnitude. A noticeable large spike after 500 seconds is visible signifying the covariance inflation after maneuver detection. Similar to the positional states of scenario 2, several large spikes after *1000* seconds occur indicating multiple runs in which the end maneuver detection happened at several times.

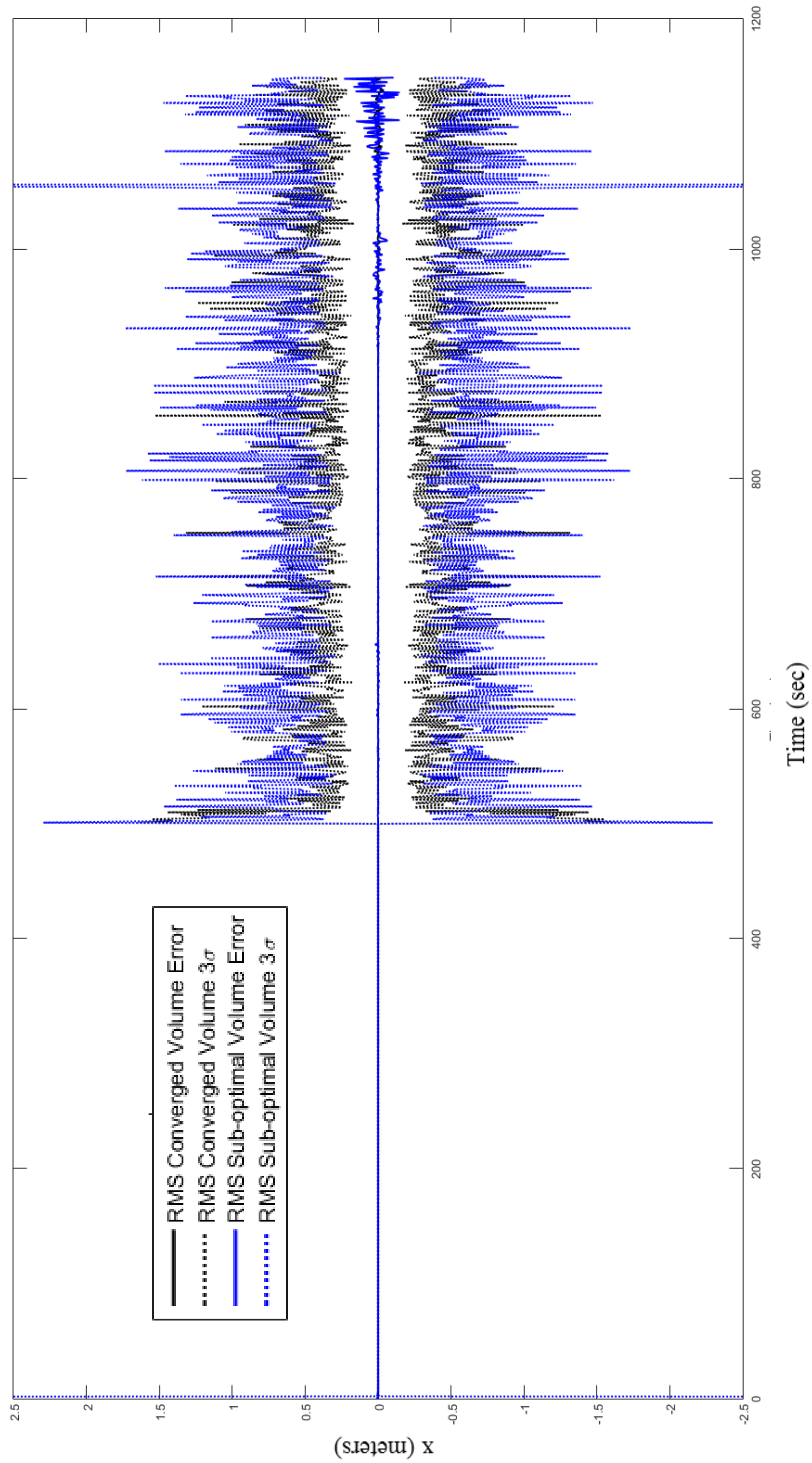

Figure **4-17**: Scenario 1 RMS residual error of the **x** velocity state with 3σ covariance bounds

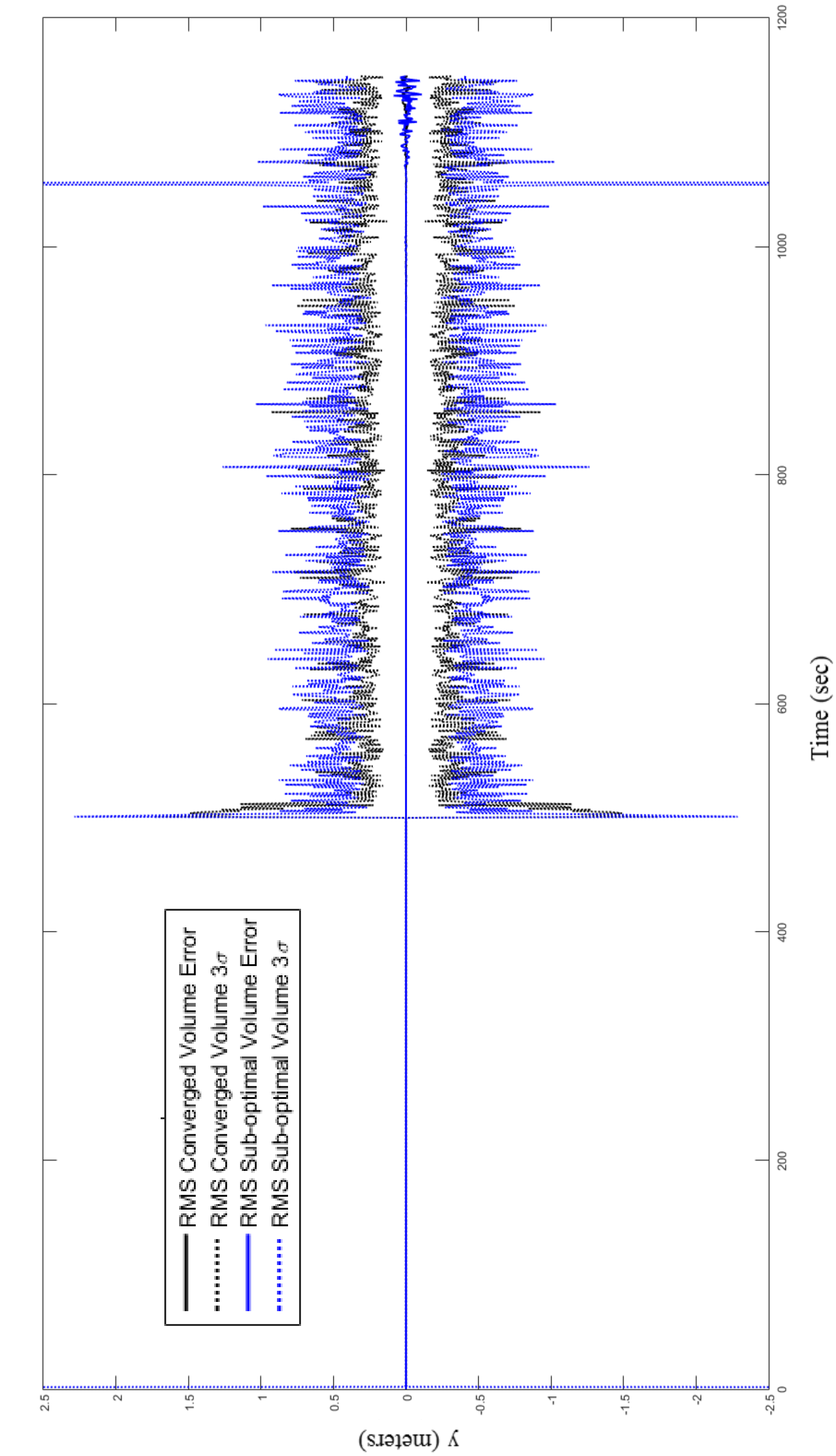

Figure **4-18**: Scenario 1 RMS residual error of the **y** velocity state with 3σ covariance bounds

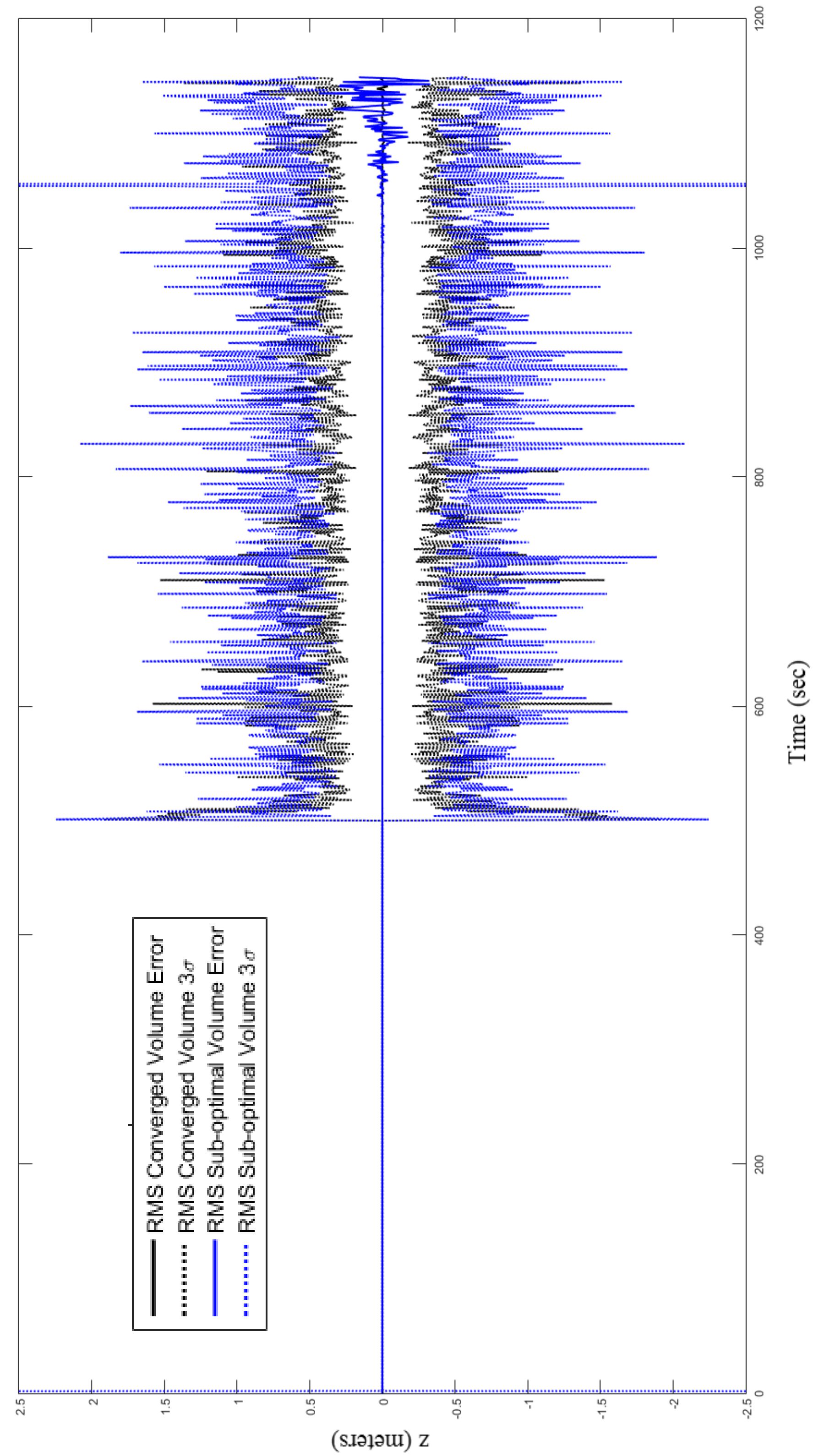

Figure **4-19**: Scenario 1 RMS residual error of the **z** velocity state with 3σ covariance bounds

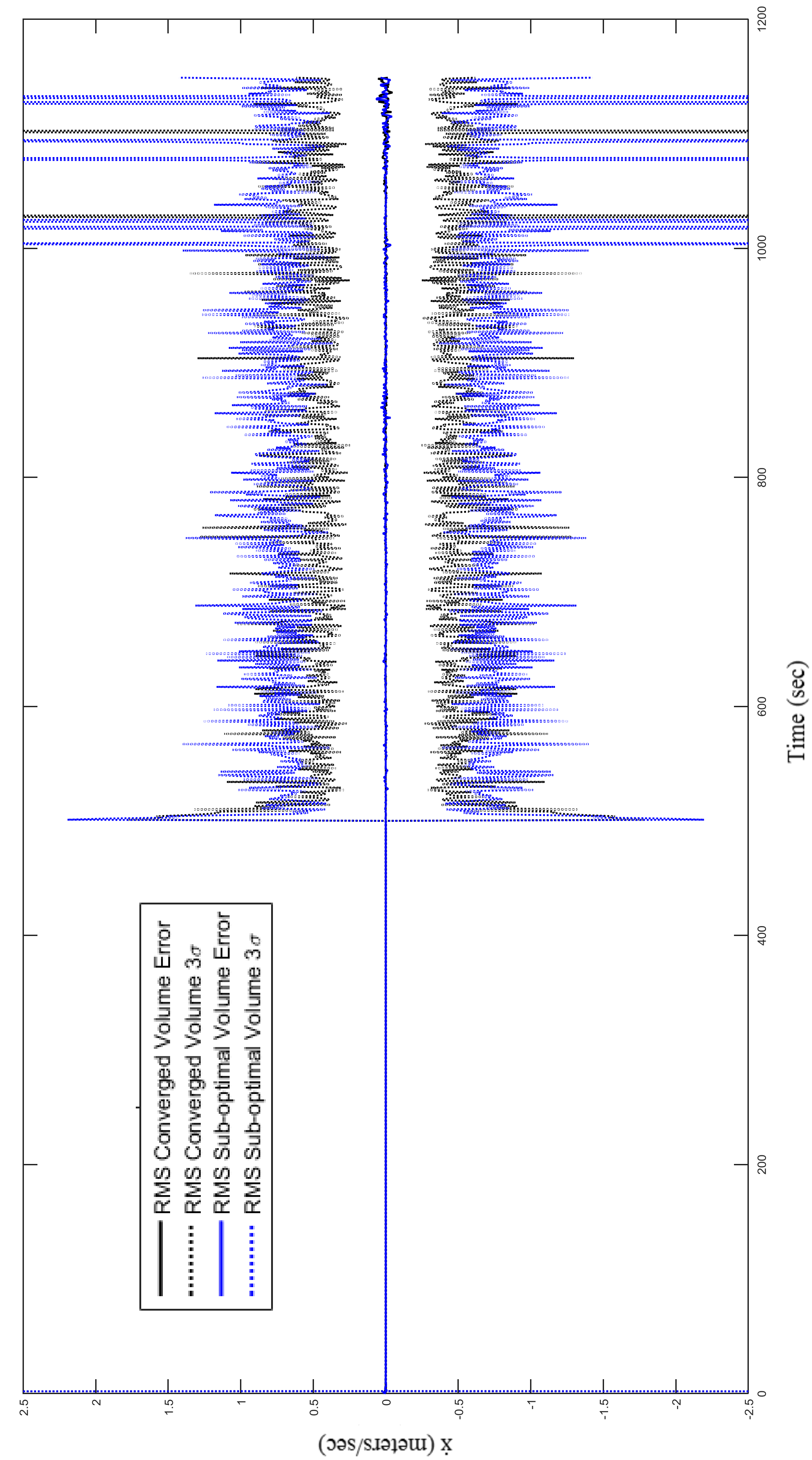

Figure **4-20**: Scenario 2 RMS residual error of the **x** velocity state with 3σ covariance bounds

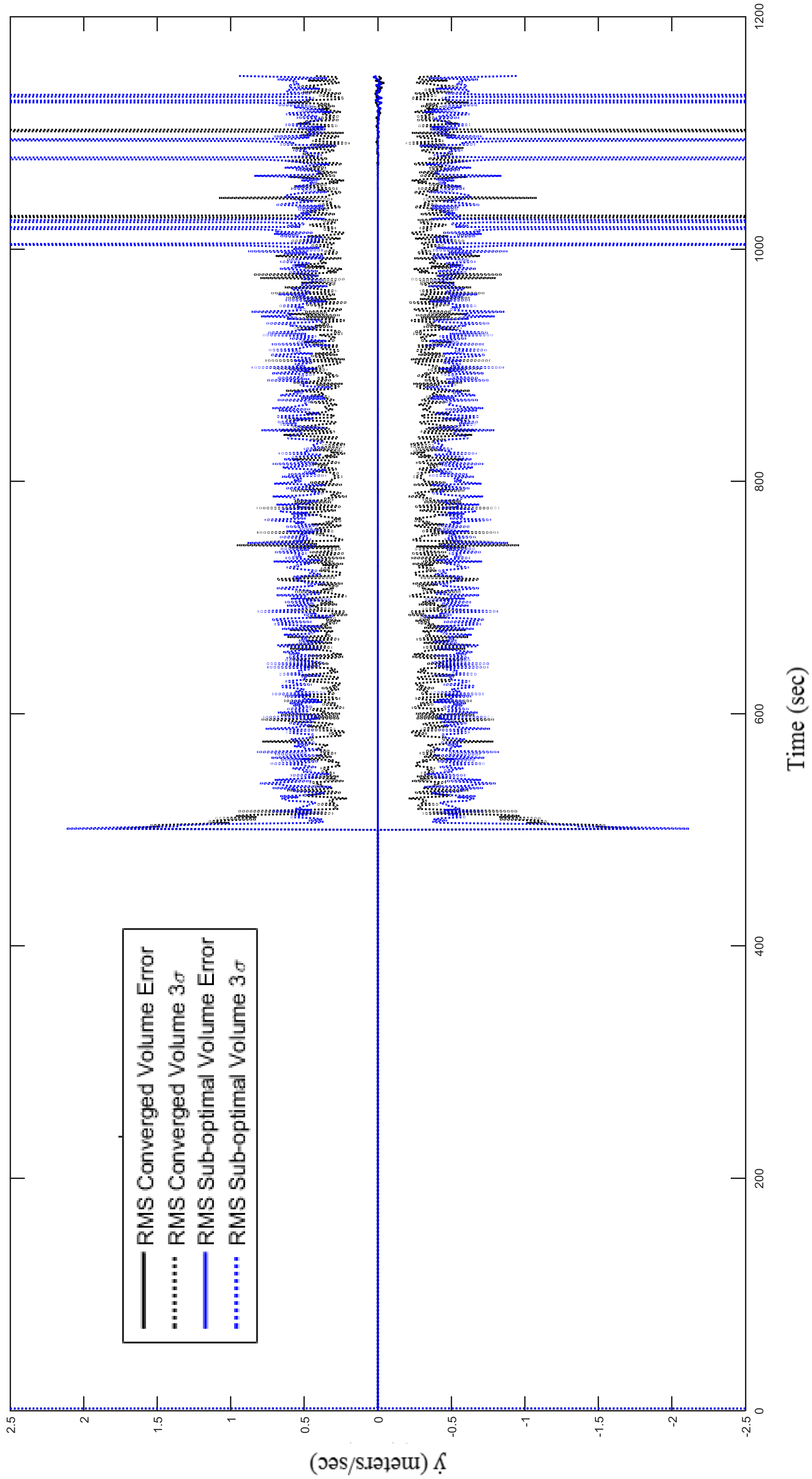

Figure **4-21**: Scenario 2 RMS residual error of the **y** velocity state with 3σ covariance bounds

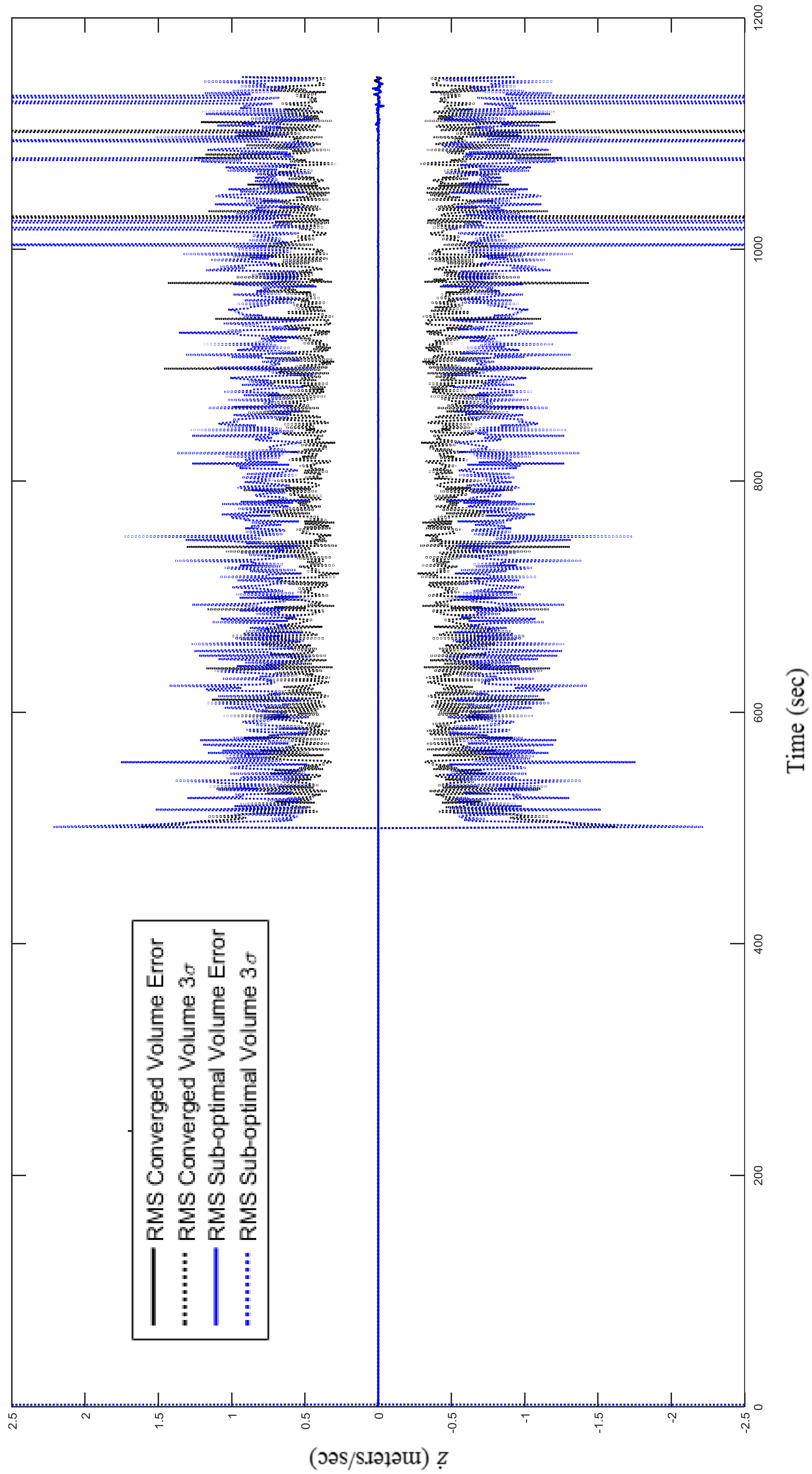

Figure **4-22**: Scenario 2 RMS residual error of the **z** velocity state with 3σ covariance bounds

4.3.2. Attitude Estimation

Attitude estimation represents the rotation from the LVLH frame to the observer spacecraft body frame. Attitude is represented in the quaternion formulation though all subsequent plots are converted to CRP as the corresponding covariance states in the filter were stored using the CRP attitude representation as described in the MEKF formulation. Since the attitude is the only controlled state in this dissertation, it is subject to not only the observations but the resulting guidance and control. Lighting conditions have a great effect on the quality of the observations and the resulting guidance and control. When conditions are especially dark, the control can be susceptible to larger attitude errors which are reflected in the results. Lighting becomes less optimal as the simulation progresses causing an additional source of error in the control law.

The attitude estimation is stable as shown in Figures **4-23** and **4-24** for both scenarios before the maneuver occurs. After the maneuver occurs in scenario 1 (Figure **4-23**), the attitude error begins to grow for the case in which the octree target mapping is incomplete. It should be noted that while the error increases, the target satellite is still within view though not in the center of the image frame. As the maneuver approaches its end time, the sub-optimal octree mapping attitude estimate begins to realign itself with the true attitude. By the end of the simulation, both the robust and sub-optimal octree target mapping are slightly offset from the true attitude. Since at this point the lighting conditions are worse than the beginning of the simulation, the attitude is no longer keeping the target in the center of the image frame.

In scenario 2 (Figure **4-24**), both octree mapping cases are very stable until just before the maneuver ends. Both attitude estimates act in a similar manner with the octree mapping having very little affect upon the resulting attitude. The lighting conditions adversely affect the attitude just as it occurred in scenario 1.

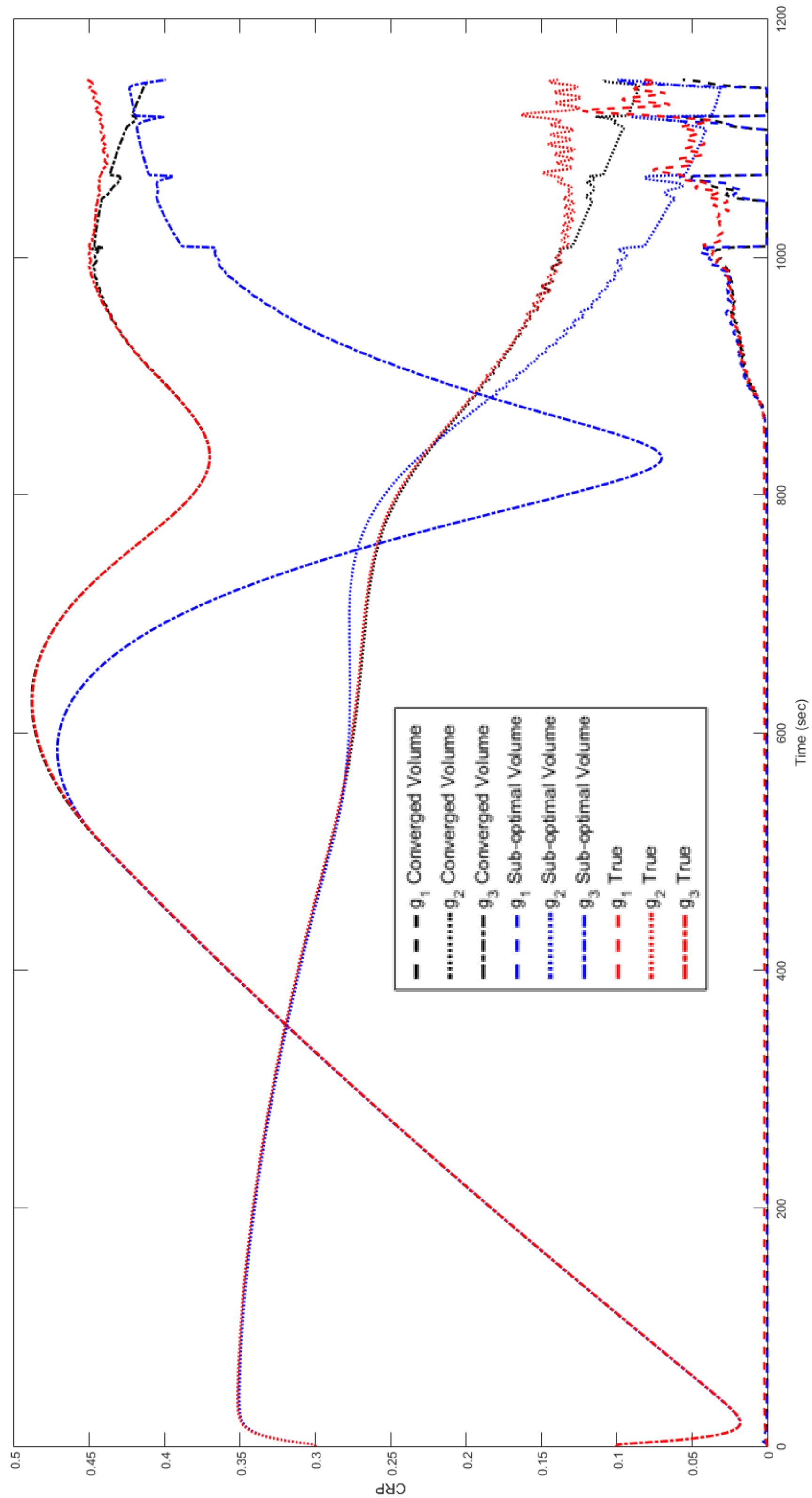

Figure **4-23**: Scenario 1 CRP attitude representation versus time

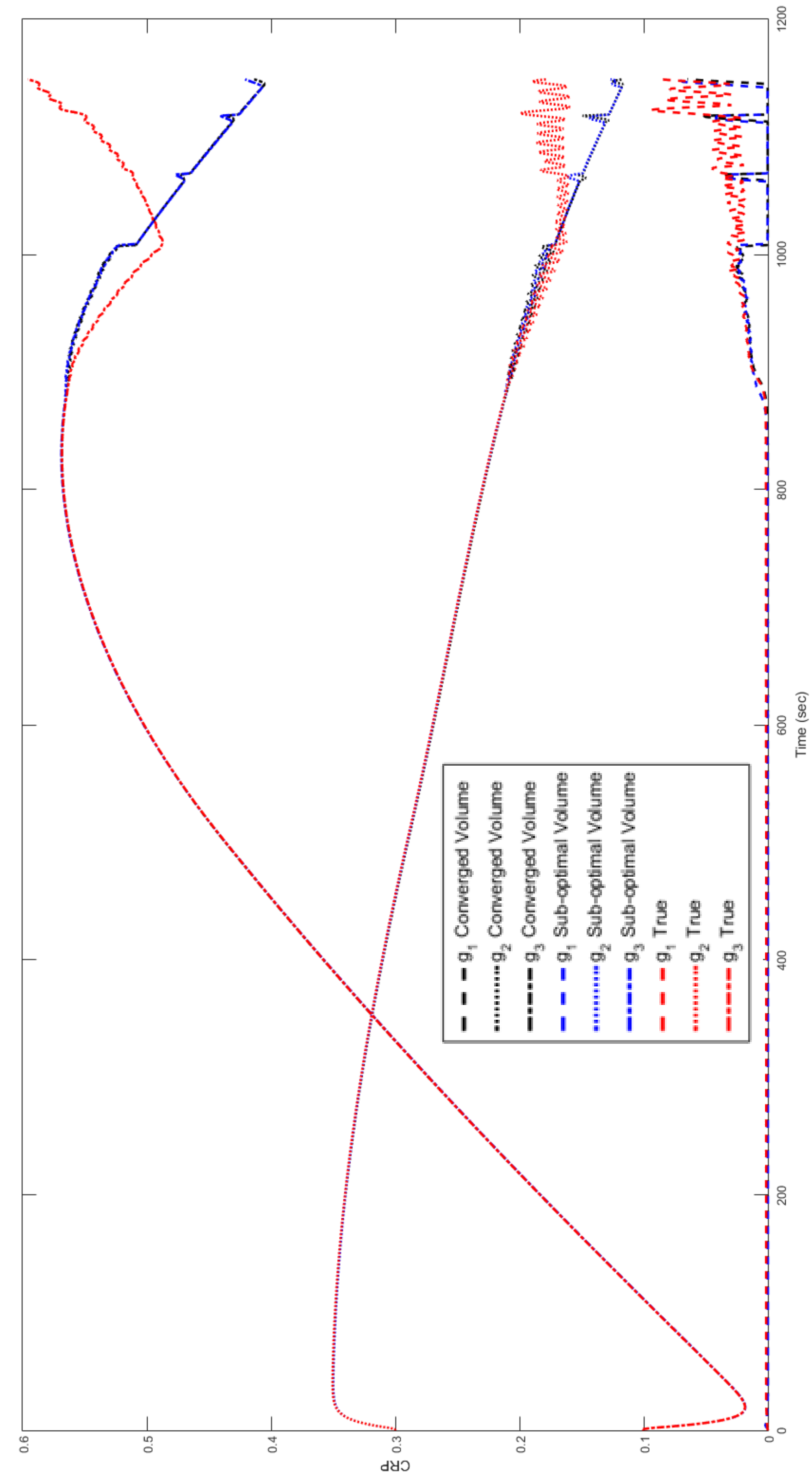

Figure **4-24**: Scenario 2 CRP attitude representation versus time

Angular velocity measurements are provided by onboard gyros providing updates with each new image. Bias state estimates in Figure **4-25** shows excellent convergence. As these estimates are independent of the other states, the gyro bias is unaffected by the uncertainty of maneuvers. They are also independent of the octree mapping so only the bias estimates from the robust octree mapping case are plotted. Angular velocities are recovered using the quaternion estimates since angular velocities were not estimated directly. In scenario 1, Figures **4-26** to **4-28** show very stable angular velocity estimates (ω_1, ω_2, ω_3) up until after the maneuver end detection occurs. These plots show the RMS residual angular velocity error from just before the maneuver occurs until the simulation end. Corresponding to the covariance inflation spike detecting the end of the maneuver is a large residual error especially in the sub-optimal octree target mapping. The CRP estimates in Figures **4-29** to **4-31** have their highest error correlating to the largest errors in the angular velocity. Residuals in the attitude oscillate in this period with the sub-optimal octree target mapping estimates having higher error on average. Two spikes in the covariance following *1000* seconds correspond to similar behavior in the scenario 1 translational states. These represent two times frequently in which the end maneuver detection occurred.

Gyro bias estimates in scenario 2 Figure **4-32** were very similar to those of scenario 1, which is expected behavior. The angular velocity RMS residual errors for scenario 2 in Figures **4-33** to **4-35** reflect similar behavior to that exhibited in scenario 1. However, the variance in the residual error is less than those present in scenario 1 for both octree mapping cases. As in scenario 1, the robust octree mapping residual are better on average though with less disparity. RMS residual error of the attitude in scenario 2 (Figures **4-36** to **4-38**) follows the same trends as scenario 1. Similar to the translational states of scenario 2, multiple covariance spikes are present attributed to the fact that detecting the end of the maneuver is more difficult with lower thrust magnitude. The error for both sub-optimal and robust octree mapping cases is relatively the same with no inherent advantage for the more optimal mapping.

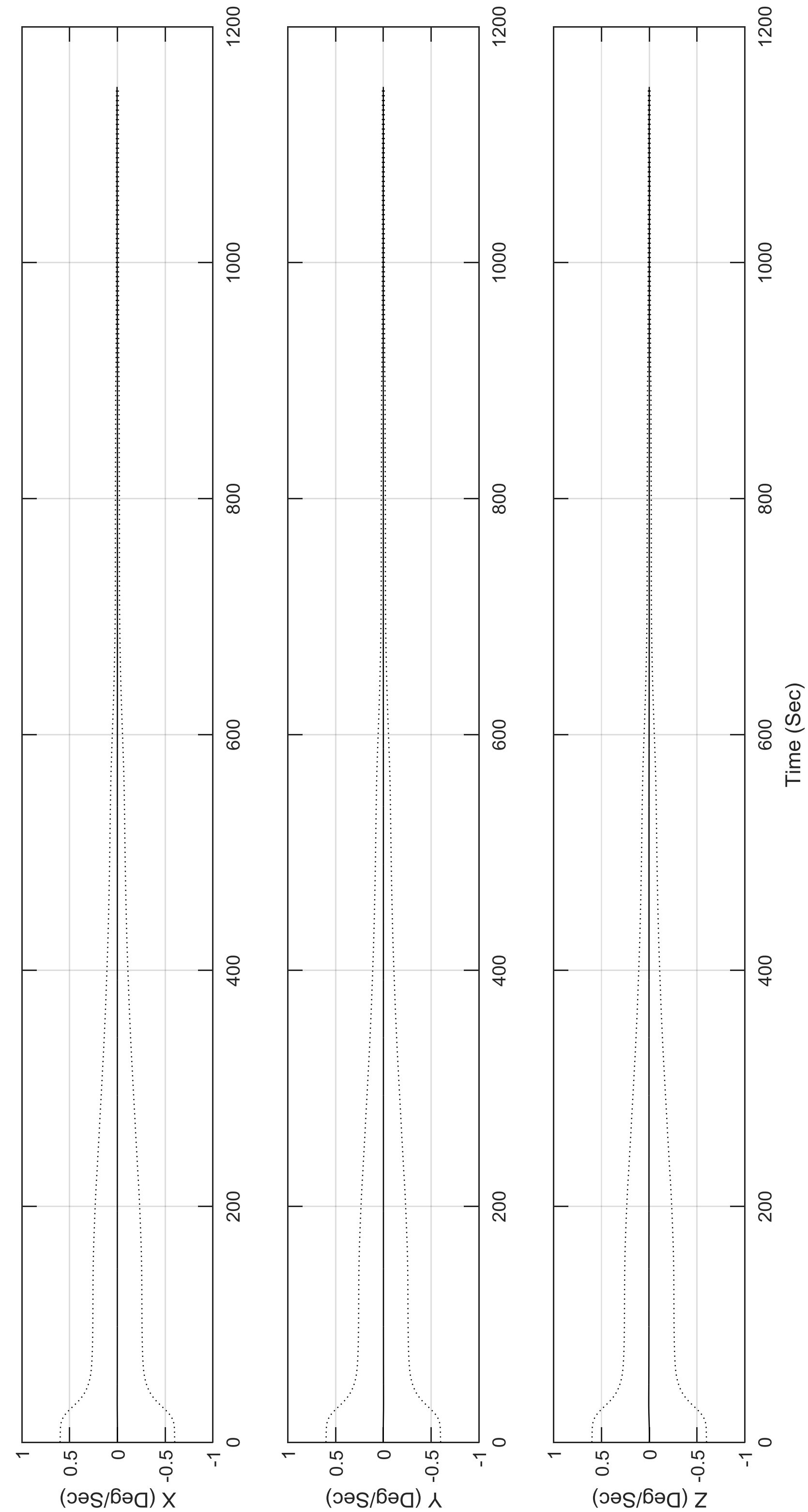

Figure **4-25**: Scenario 1 gyro bias estimates

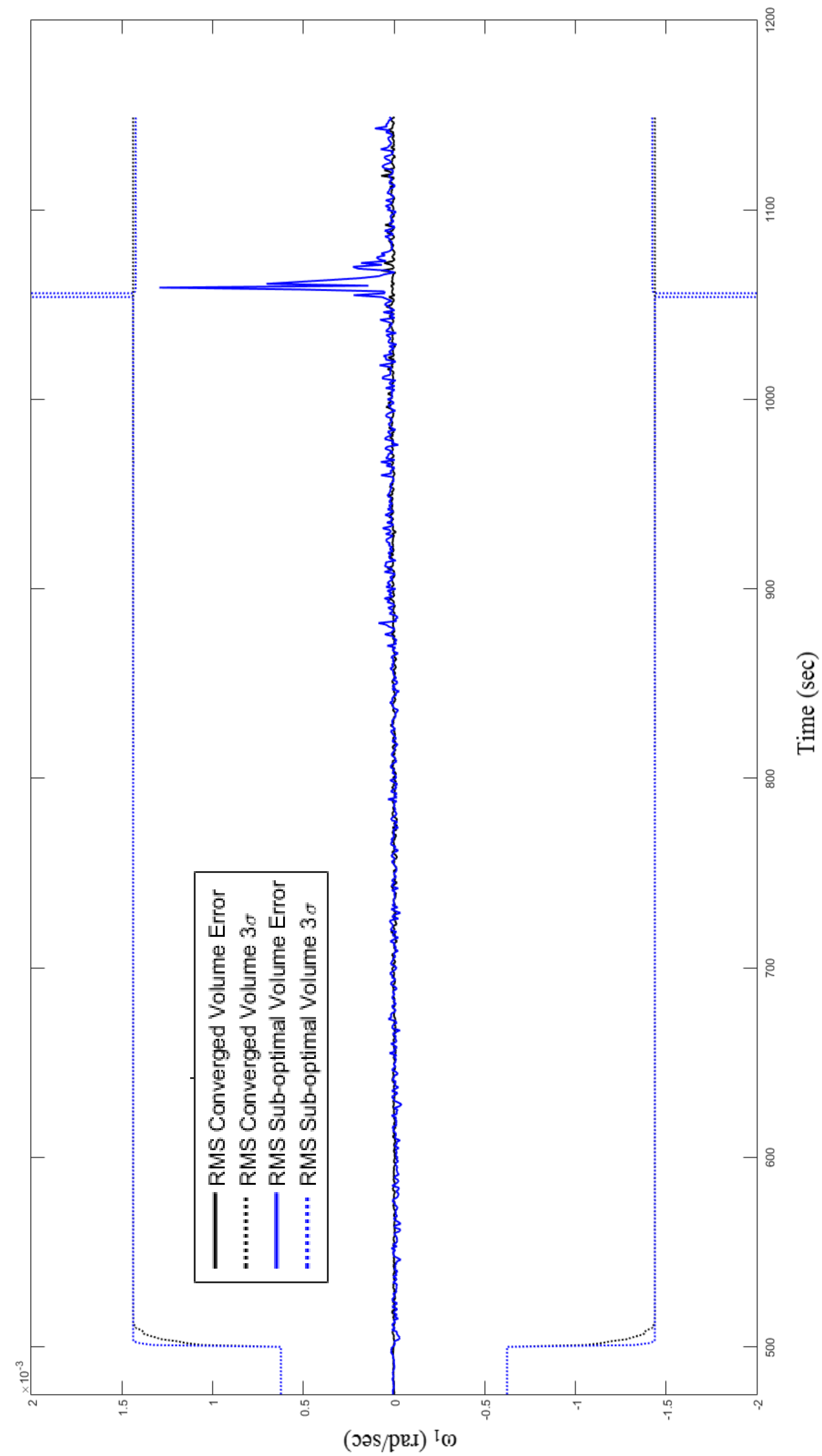

Figure **4-26**: Scenario 1 angular velocity in **x**

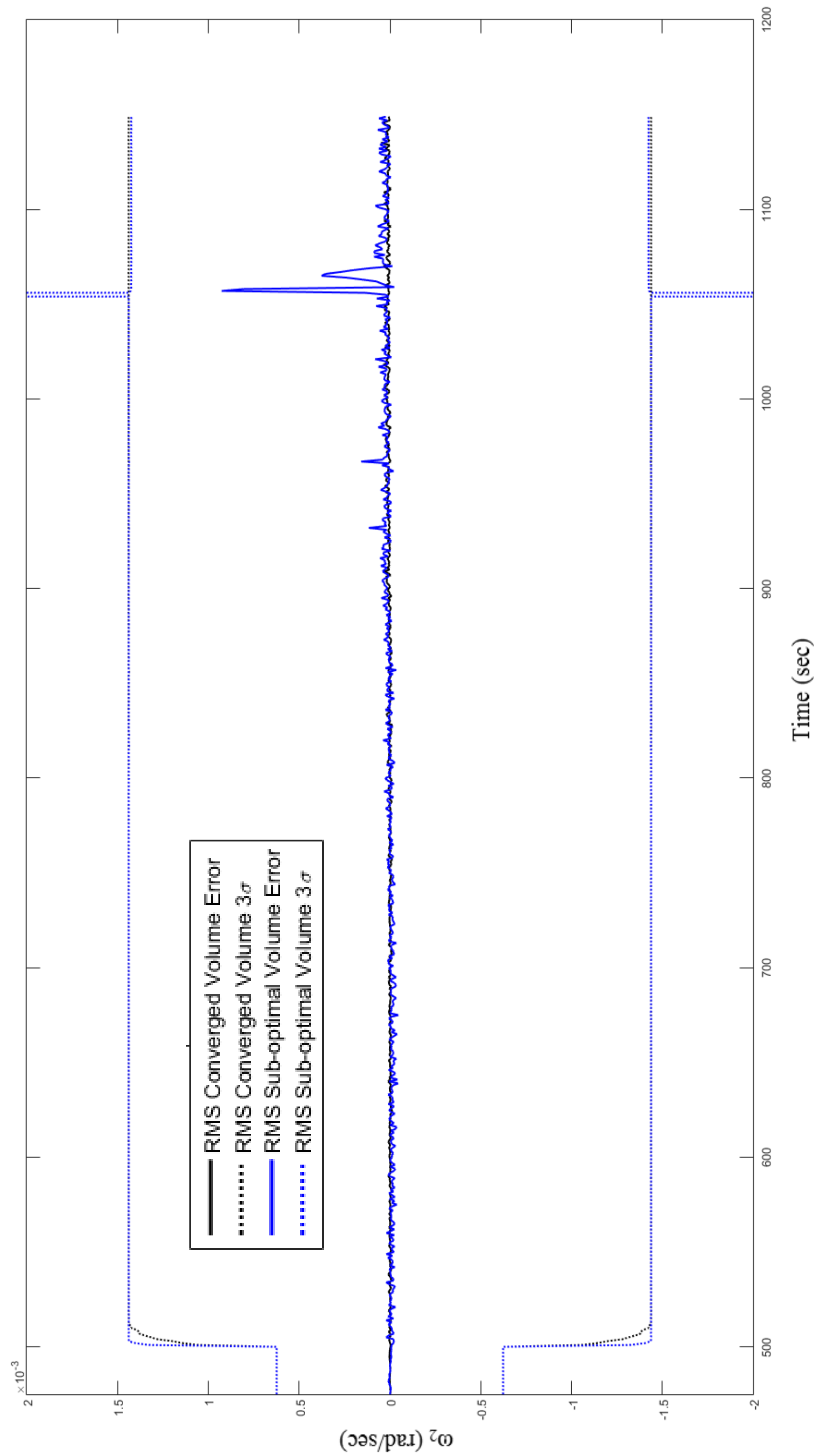

Figure **4-27**: Scenario 1 angular velocity in **y**

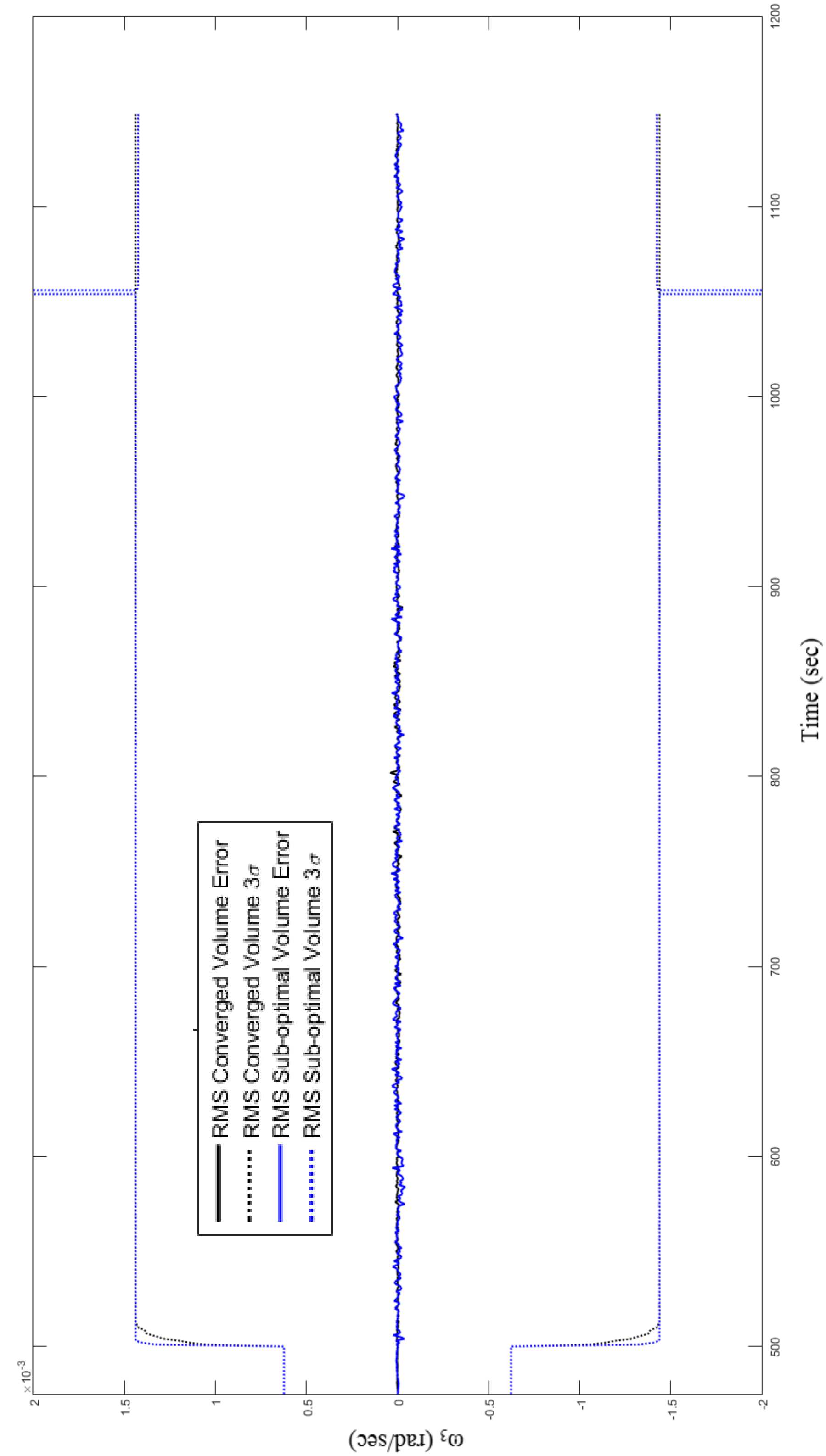

Figure **4-28**: Scenario 1 angular velocity in **z**

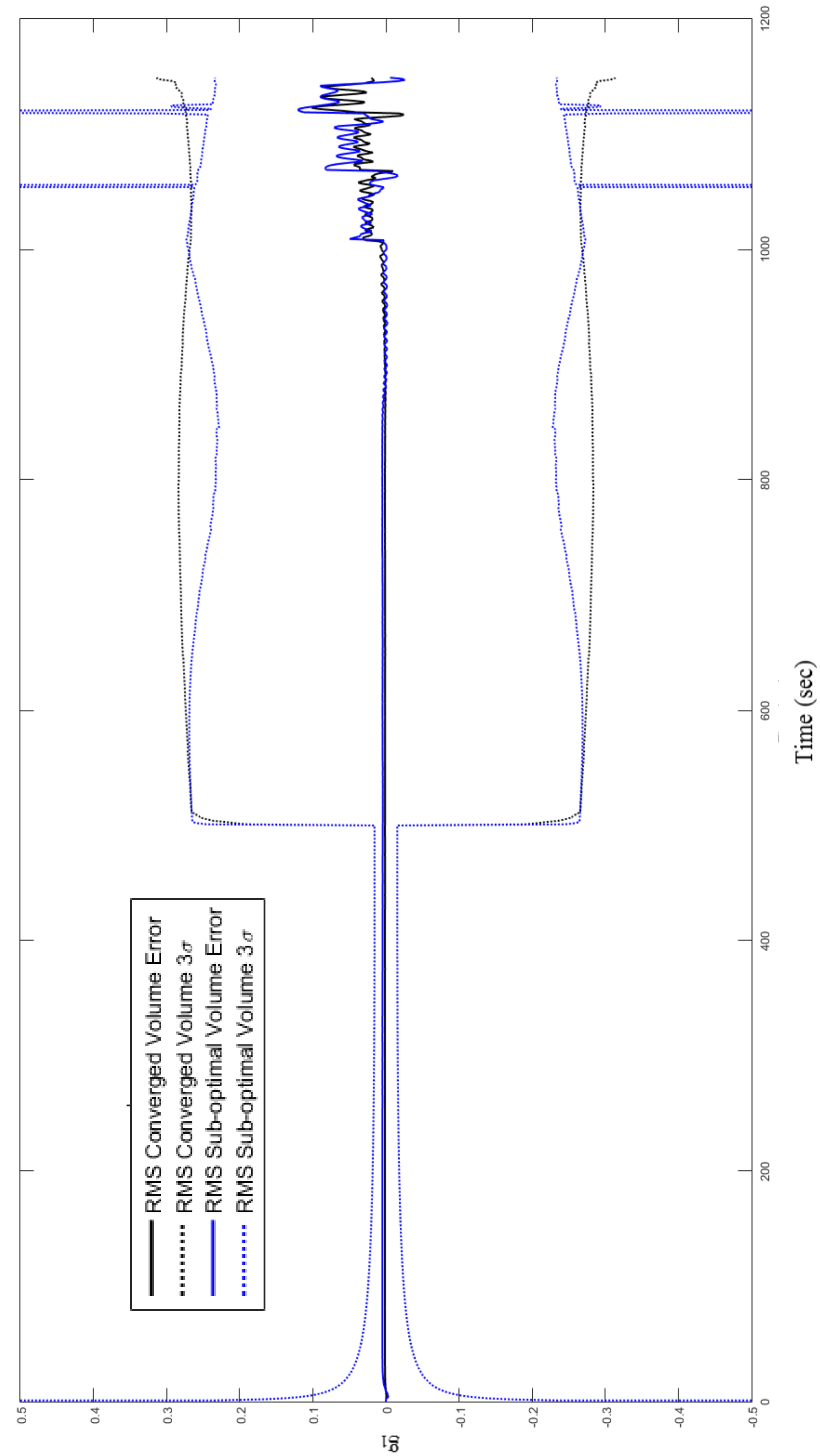

Figure **4-29**: Scenario 1 RMS residual error of the $\mathbf{g_1}$ CRP state with 3σ covariance bounds

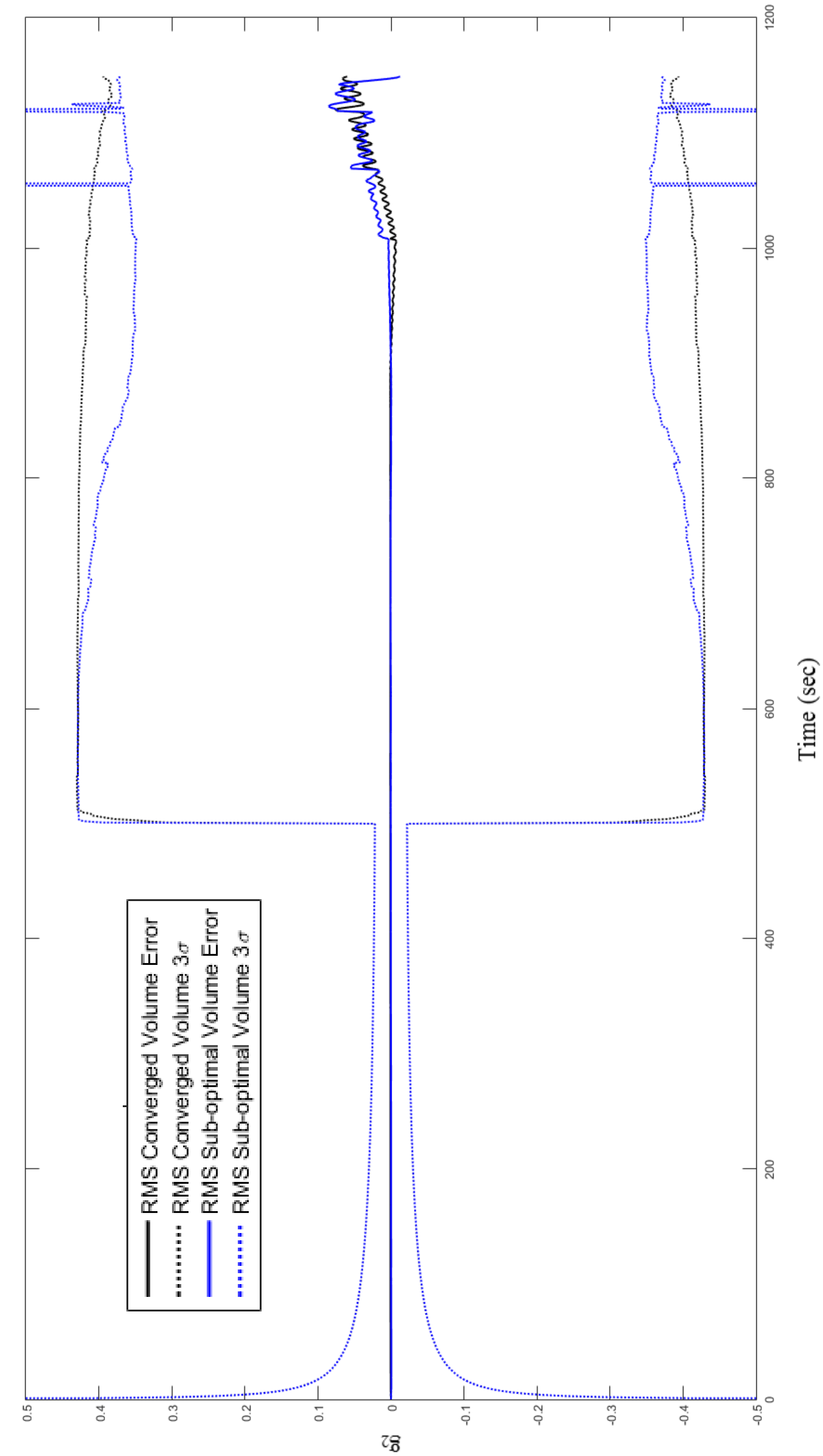

Figure **4-30**: Scenario 1 RMS residual error of the $\mathbf{g_2}$ CRP state with 3σ covariance bounds

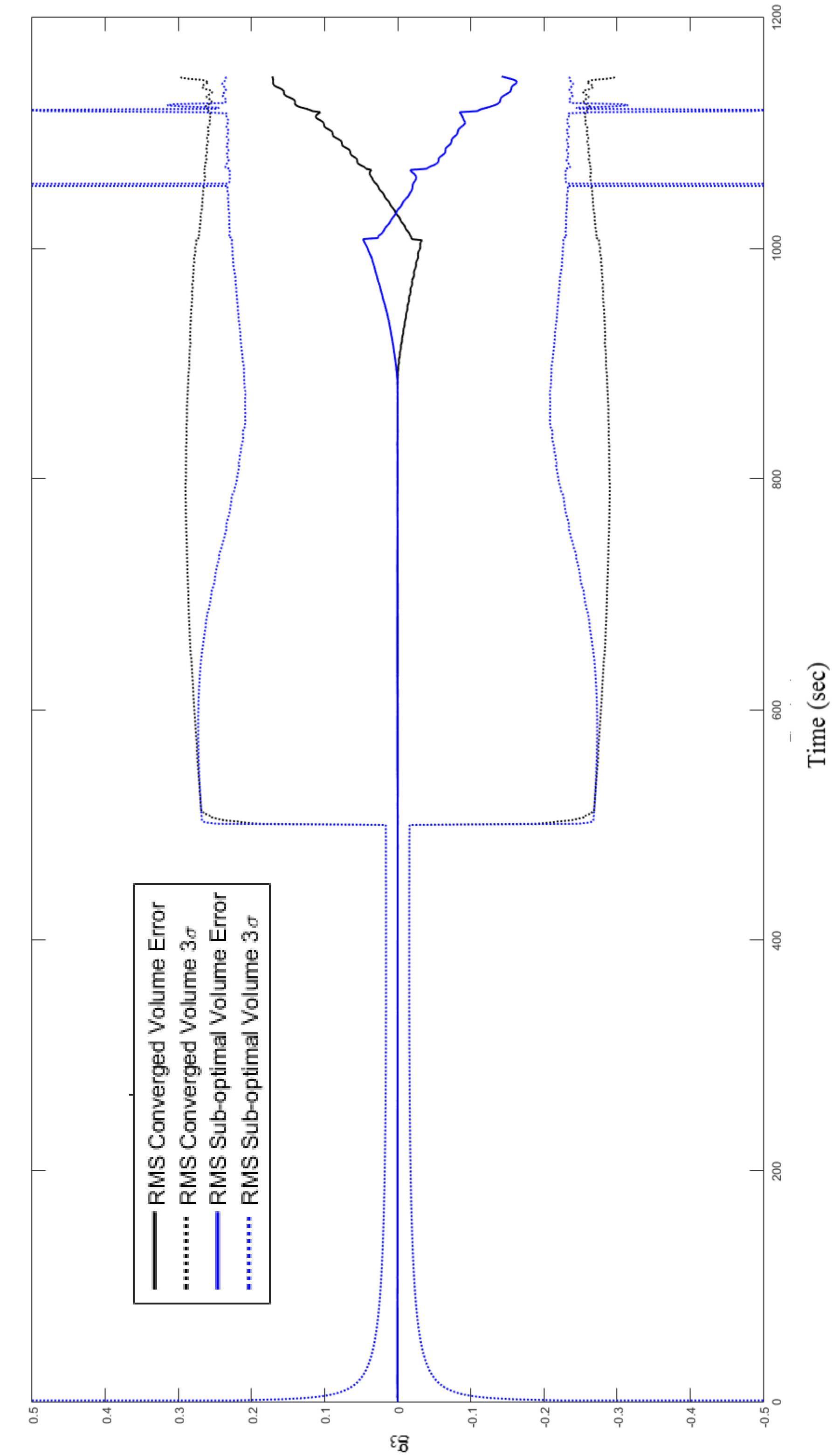

Figure **4-31**: Scenario 1 RMS residual error of the $\mathbf{g_3}$ CRP state with 3σ covariance bounds

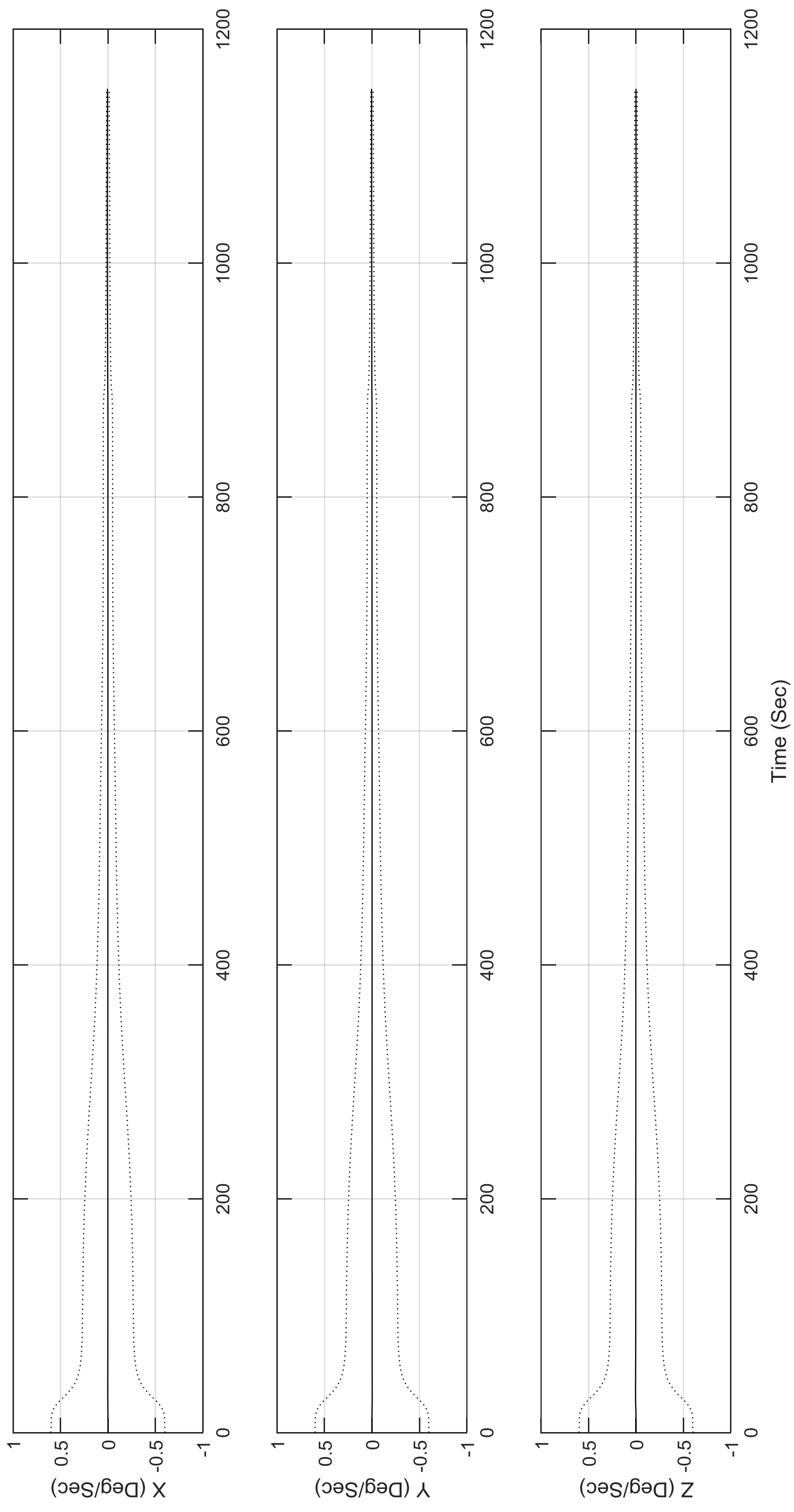

Figure **4-32**: Scenario 2 gyro bias estimates

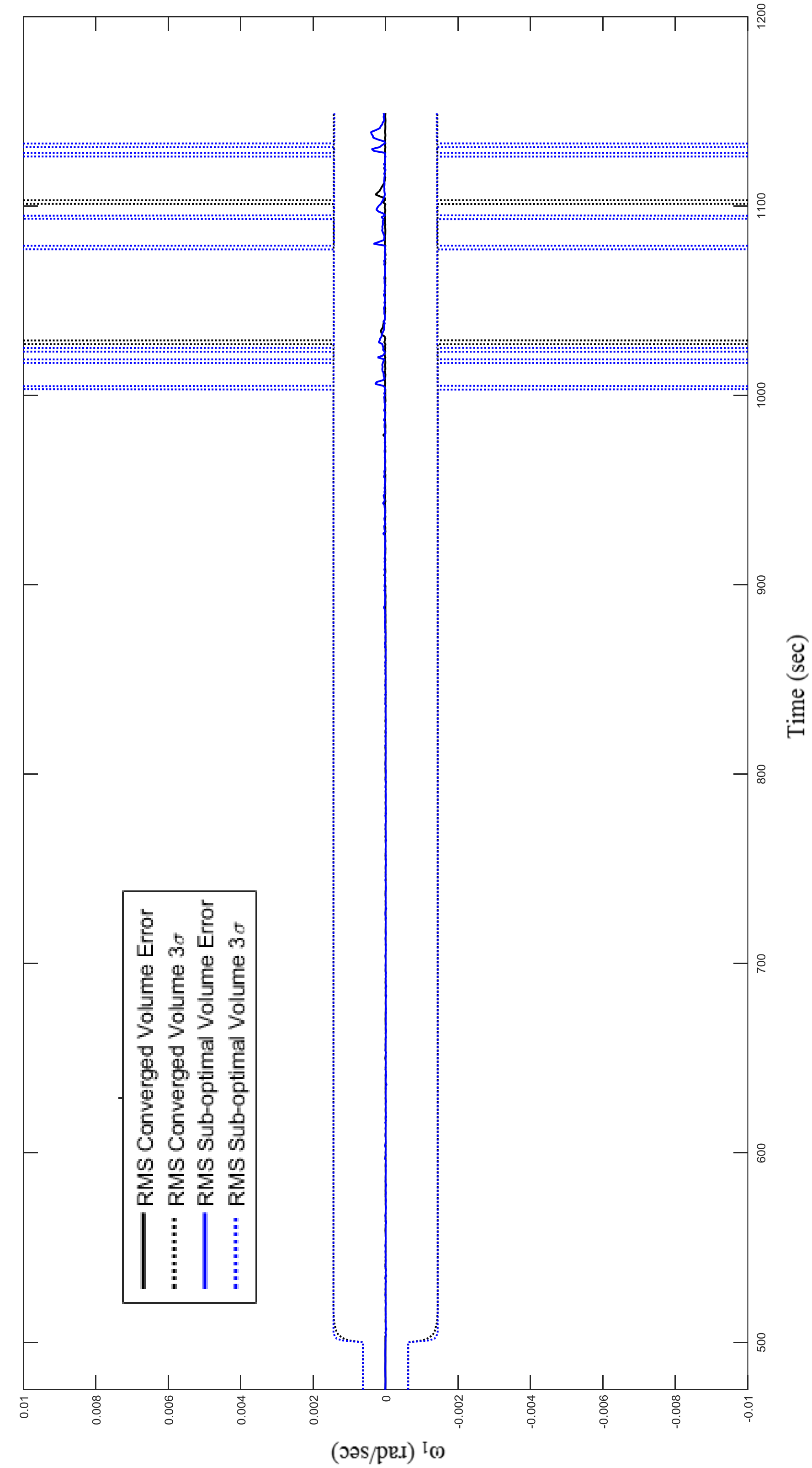

Figure **4-33**: Scenario 2 angular velocity in **x**

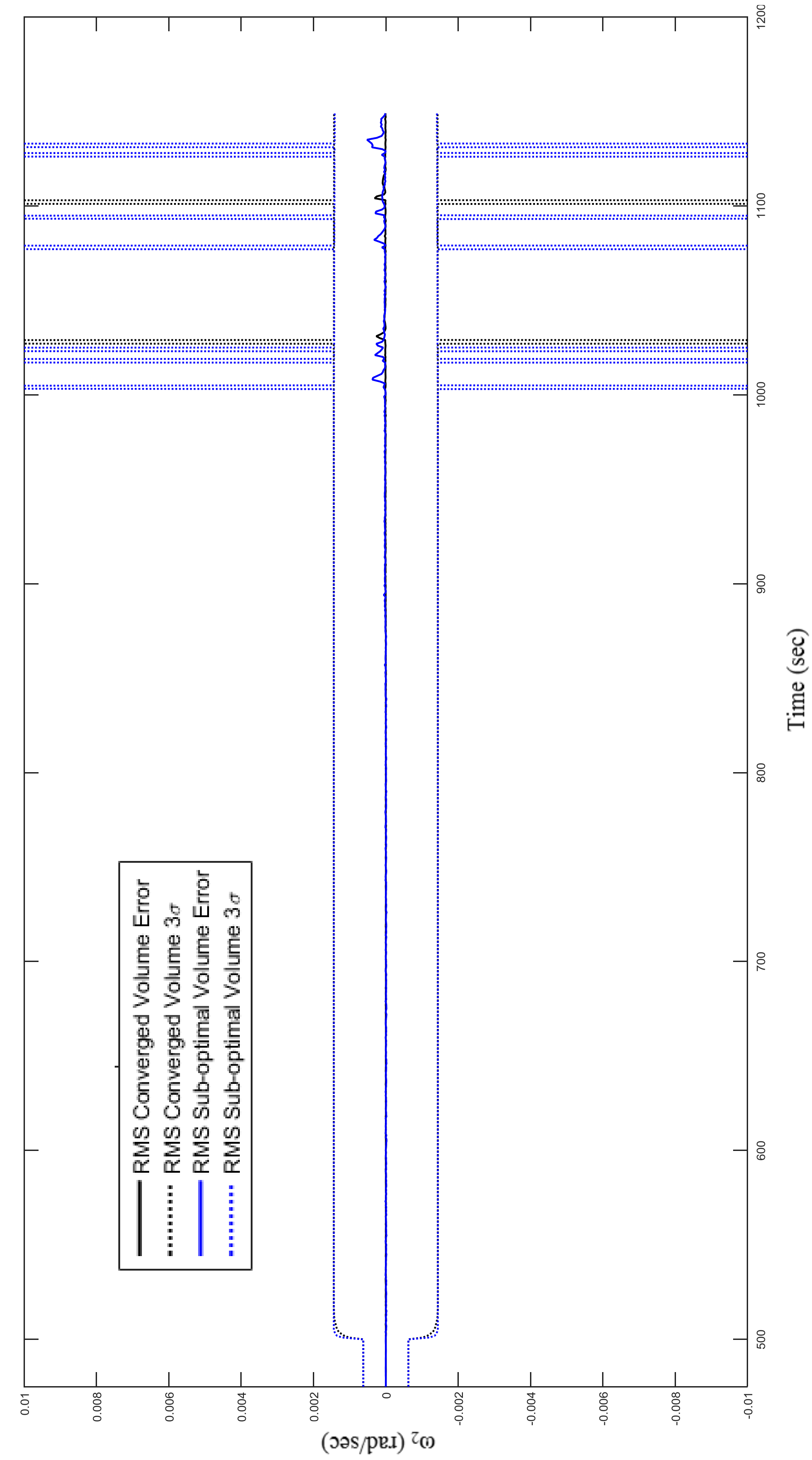

Figure **4-34**: Scenario 2 angular velocity in **y**

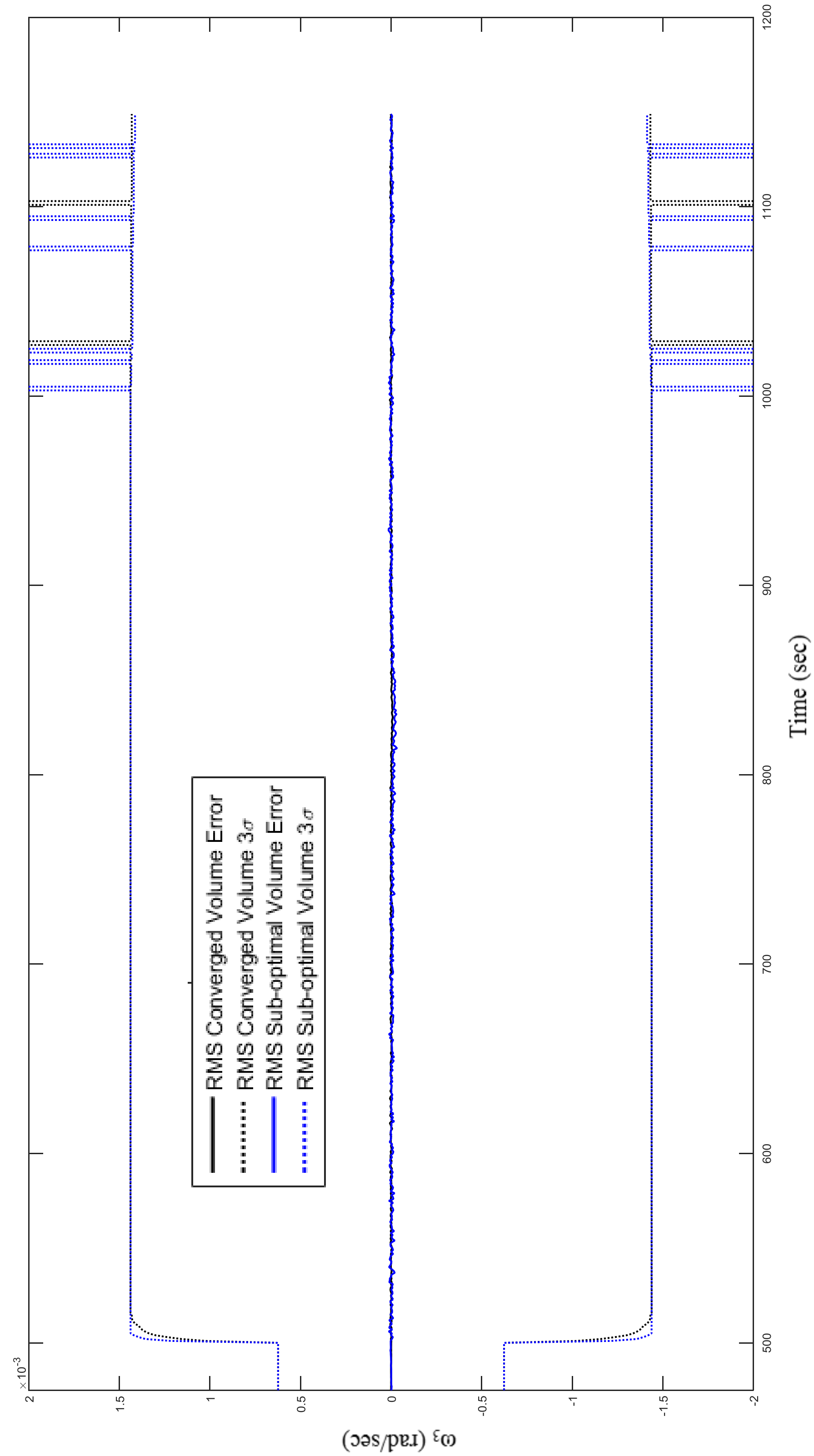

Figure **4-35**: Scenario 2 angular velocity in **z**

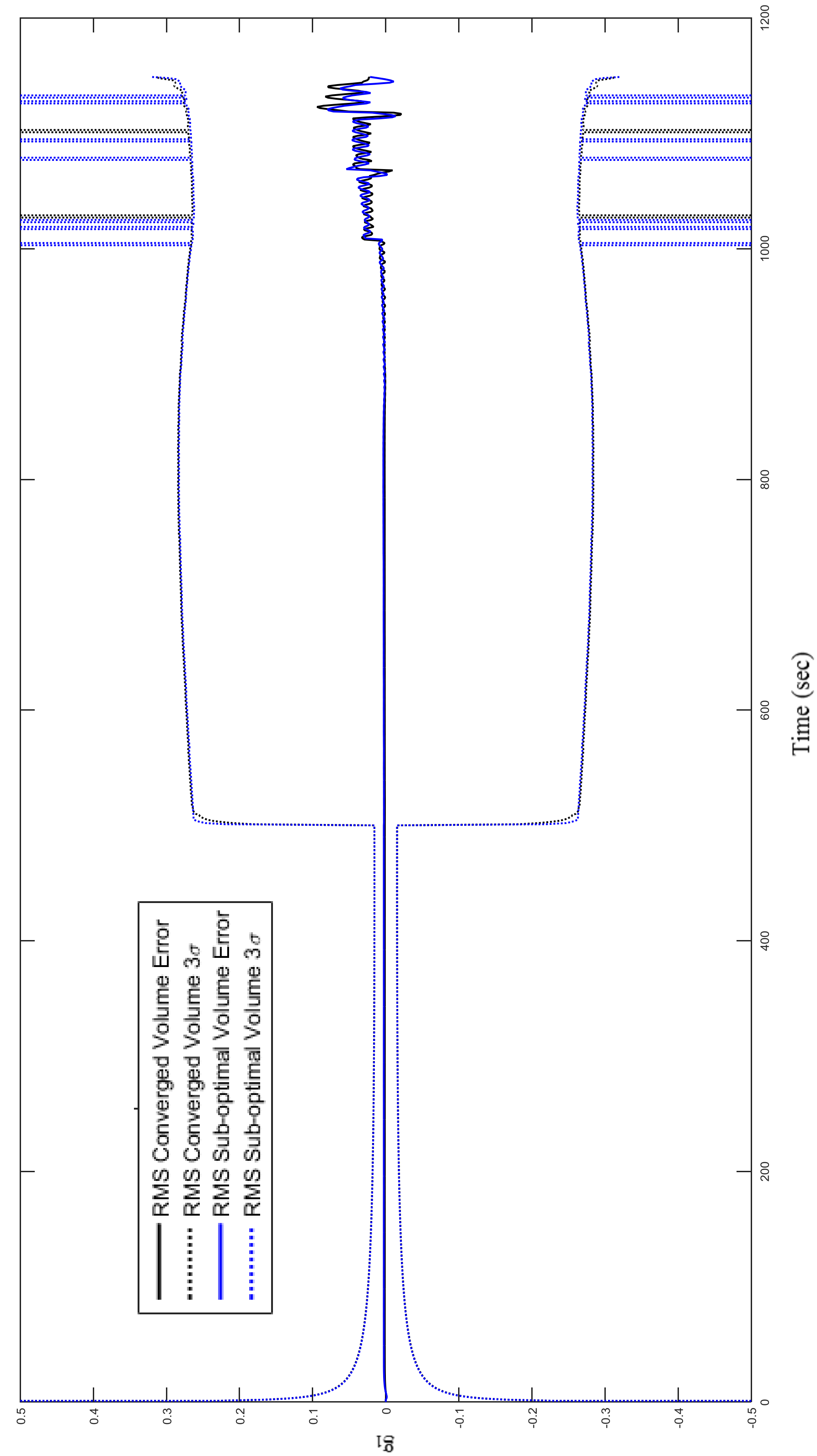

Figure **4-36**: Scenario 2 RMS residual error of the $\mathbf{g_1}$ CRP state with 3σ covariance bounds

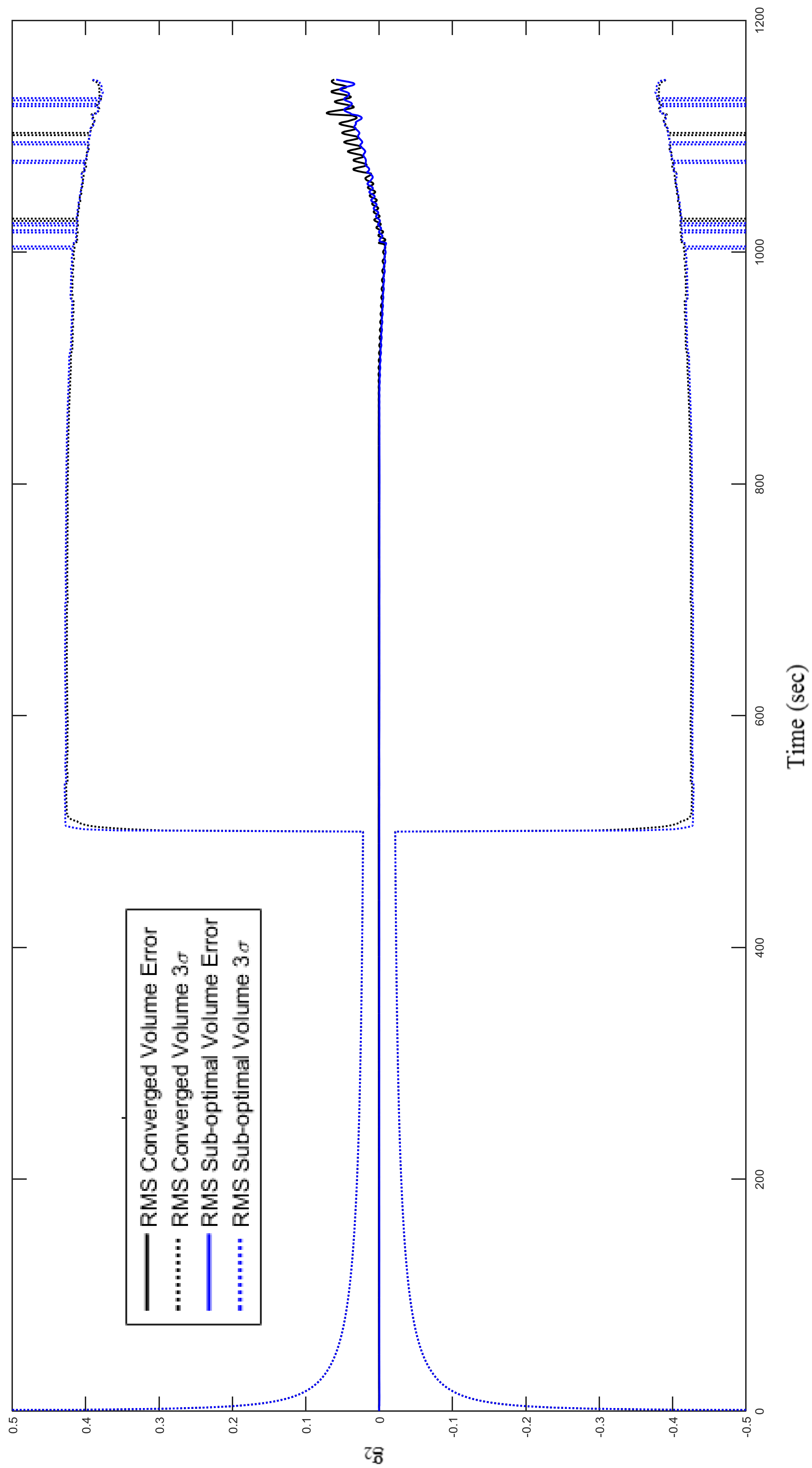

Figure **4-37**: Scenario 2 RMS residual error of the $\mathbf{g}_2$ CRP state with 3σ covariance bounds

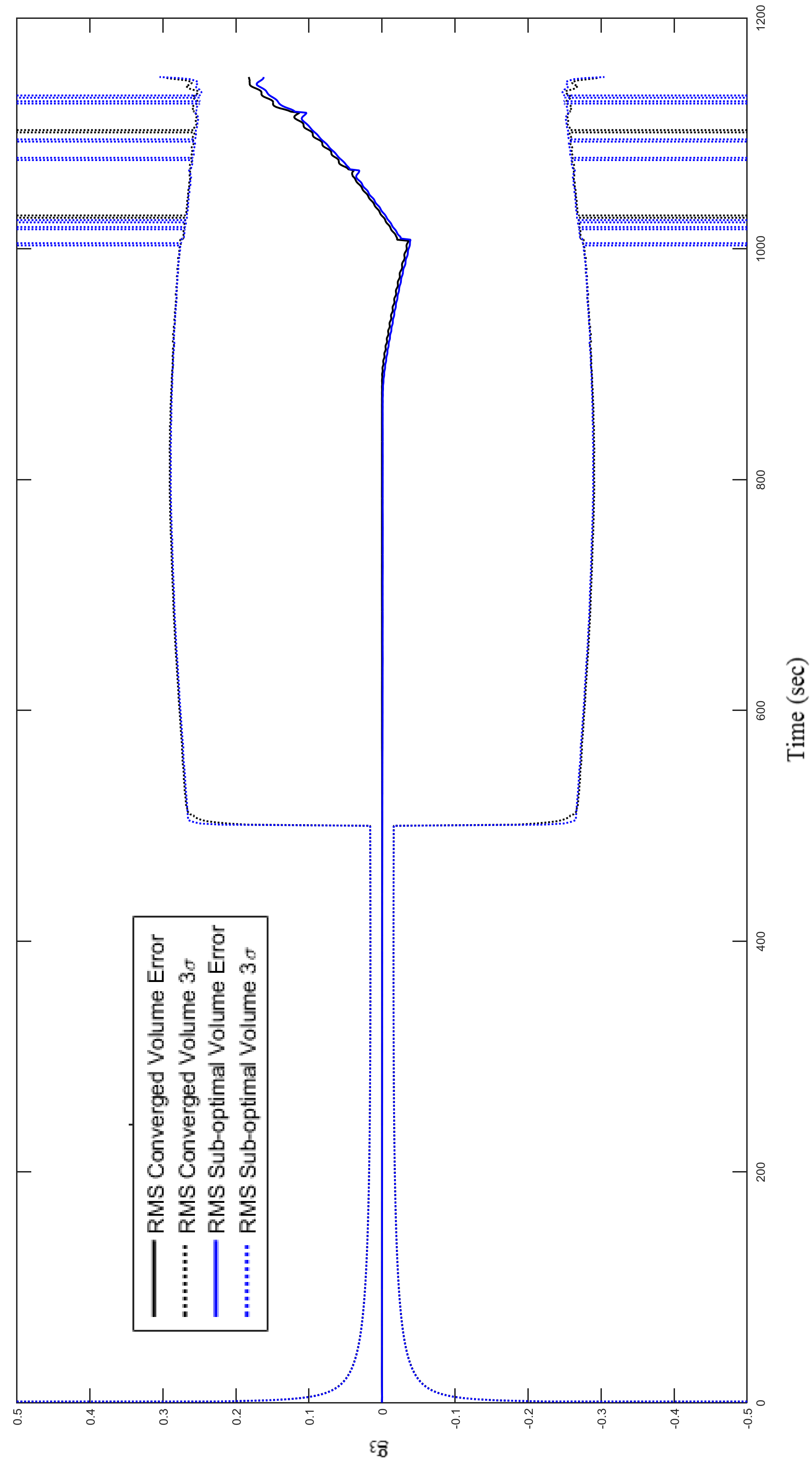

Figure **4-38**: Scenario 2 RMS residual error of the $\mathbf{g_3}$ CRP state with 3σ covariance bounds

4.3.3. Thrust Estimation and Maneuver Detection

Detecting a maneuver is highly dependent upon several factors. The overall noise inherent in a system, imperfect model dynamics, and measurement noise can either mask a maneuver or create a false positive. These are the contributing reasons as to why a continuous thrusting maneuver is easier to detect than an impulsive thrust. A continuous maneuver usually lasts for longer durations (though at lower thrust) such that an observer tracking the object has ample time to detect inconsistencies in the motion resulting from using a non-thrusting model to estimate the translational states of the tracked object.

To avoid situations in which false maneuvers are detected or maneuvers go unnoticed, tuning the Mahalanobis distance and covariance inflation parameters correctly are vital. In choosing the Mahalanobis distance threshold, a chi-squared test is conducted to ensure that the threshold is higher than *99%* of the expected values. The covariance inflation threshold is chosen such that it adequately allows the switched model (quiescent to thrust or vice versa) to improve the state estimation.

The Mahalanobis distances for all simulations in each octree mapping case are shown in Figures **4-39** and **4-40** where the red-dashed line denotes the Mahalanobis distance threshold. In scenario 1, the left plot of Figure **4-39** shows the moment in which the maneuver is detected. Any Mahalanobis distances above the red-dashed line signify that a maneuver has been detected.
It is a very narrow region around *500* seconds into the simulation occupied by the Mahalanobis distance. This indicates that the maneuver detection occurred consistently shortly after it happened for both octree mapping cases. After the maneuver is detected, the Mahalanobis distance is reduced below the threshold. Examining Figure **4-39** from left to right, the Mahalanobis distance remains beneath the threshold until the maneuver ends. The region encompassing the maneuver detection end is considerably wider than the maneuver detection at

the start. Due to the increased uncertainty during the maneuver and inherent noise in the measurements from the diminishing lighting conditions, detecting the ending of the maneuver is more difficult. The time period in which the maneuver end detection occurs for the robust octree target mapping is between *1010* seconds and *1071* seconds.

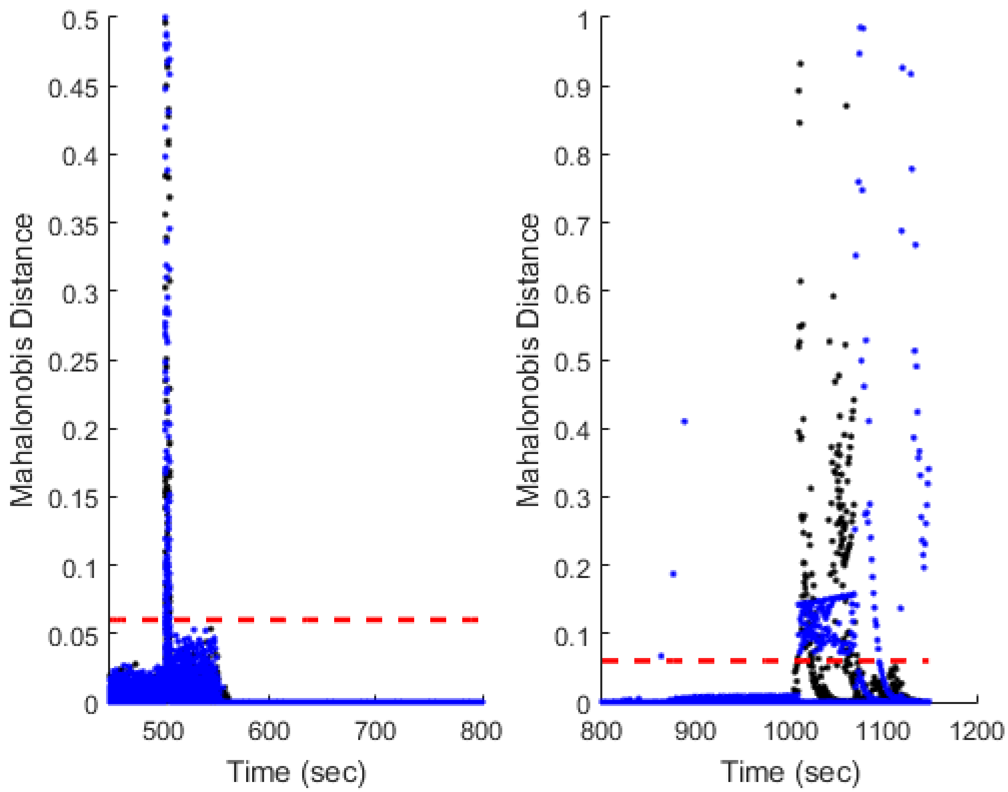

Figure **4-39:** Scenario 1 Mahalanobis distance of the robust (black) and sub-optimal (blue) octree mapping

In contrast, the sub-optimal octree target mapping has an extended region up till the scenario end. The longer it takes to detect the end of the maneuver, the larger the error grows due to using a thrusting model for a non-thrusting object. This is the cause of the larger error in the residual translational states for the sub-optimal mapping over the more robust mapping in Figures

4-11 to **4-13** and **4-17** to **4-19** for the position and velocity states respectively. It is very clear that the sub-optimal octree target mapping has a profound affect detecting the end of the maneuver.

The Mahalanobis distances in scenario 2 (Figure **4-40**) shows no discernable difference between both octree mapping cases. In comparing the left plot of Figure **4-40** to the left plot of Figure **4-39**, the range of maneuver detection times are wider in scenario 2. This behavior is expected since the thrust maneuver is ten times smaller and is inherently harder to detect. The estimation filter needs additional time to properly detect the effects of a maneuver on the target's natural motion. Detecting the end of the maneuver presents a similarity difficult problem. In scenario 2, both octree target mapping cases have the same range of time in which end maneuvers are detected.

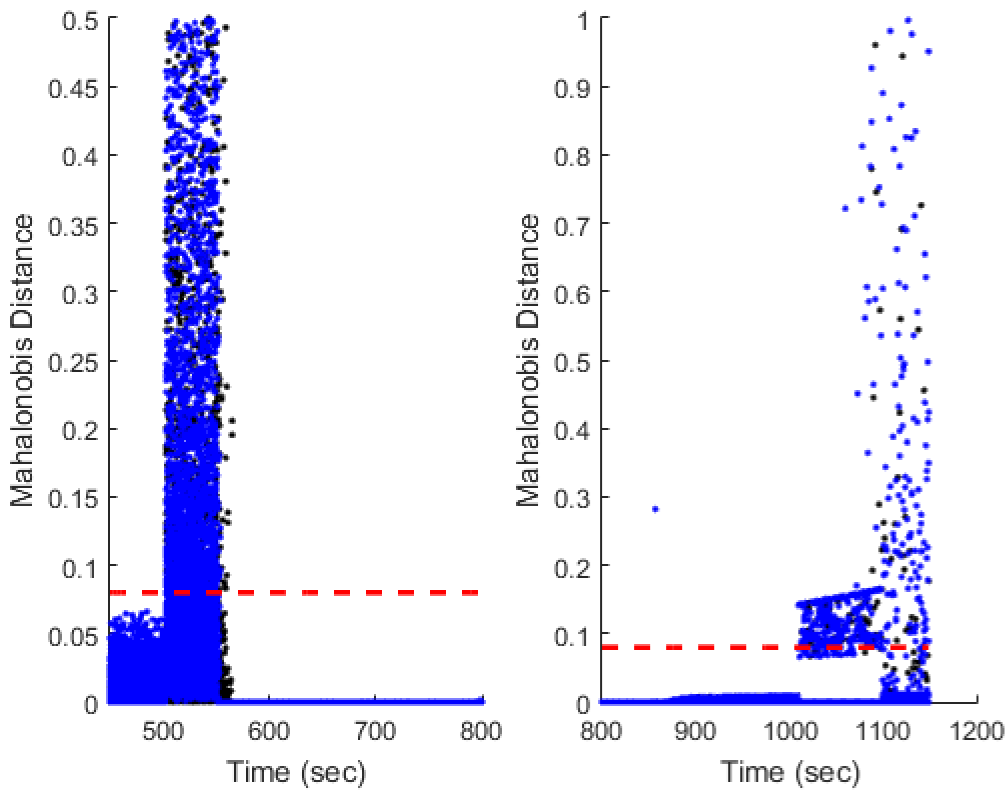

Figure **4-40:** Scenario 2 Mahalanobis distance of the robust (black) and sub-optimal (blue) octree mapping

Estimating thrust acceleration states begins after the switch from the quiescent model to the thrust model. Thrust acceleration states are appended to the state vector and are estimated during the detected maneuver until the end of the maneuver is determined. Higher thrust acceleration magnitudes are easier to estimate correctly as lower thrust acceleration magnitudes can be distorted by various factors including process and measurement noise.

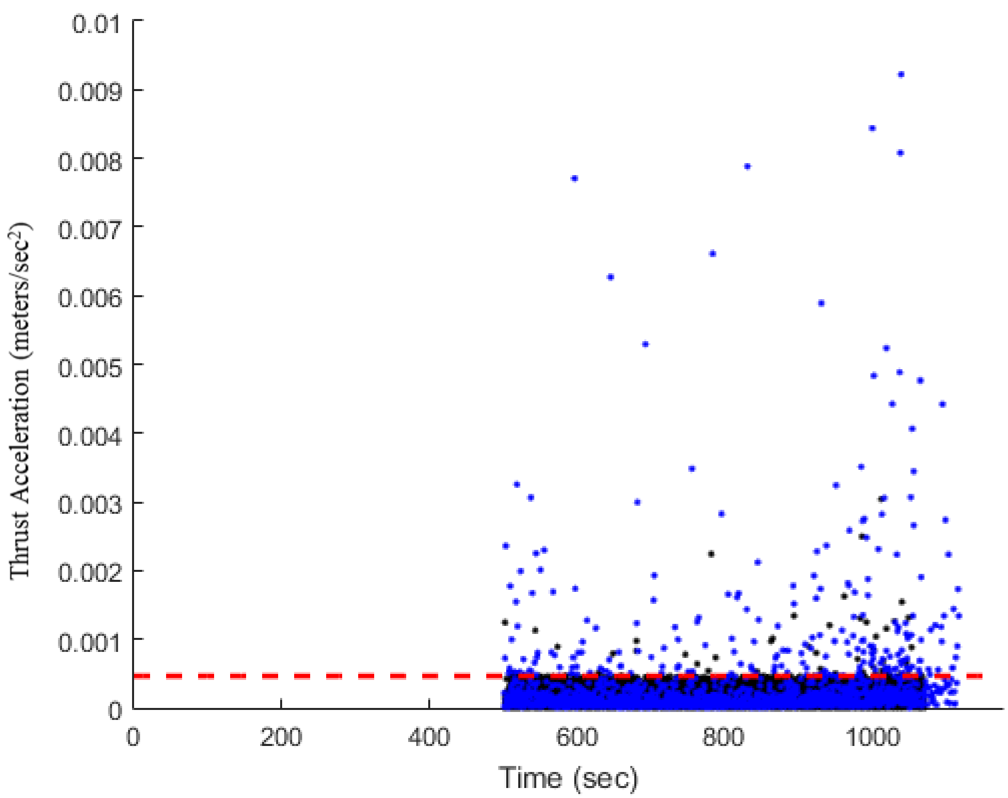

Figure **4-41:** Scenario 1 thrust magnitude estimates of the robust (black) and sub-optimal (blue) octree mapping

The thrust magnitude estimates for scenario 1 in Figure **4-41** are plotted with the true thrust acceleration magnitude (represented by the red-dashed line). Thrust acceleration magnitude estimates from the robust octree target mapping runs show a consistent convergence near the true thrust acceleration magnitude. There are several outlier estimates above the true thrust magnitude contributed from poor measurements and overall system noise. It can be seen that the distribution above the true thrust acceleration magnitudes occupies a narrower region than the sub-optimal octree target mapping. The incomplete octree target mapping has a larger variance in its estimates around the true thrust acceleration magnitude and has a wider distribution of erroneous thrust acceleration magnitude estimates above the true thrust acceleration magnitude. Furthermore, the thrust estimates in several Monte Carlo runs of the sub-optimal octree target mapping continue toward the end of the simulation time as the maneuver end failed to be detected.

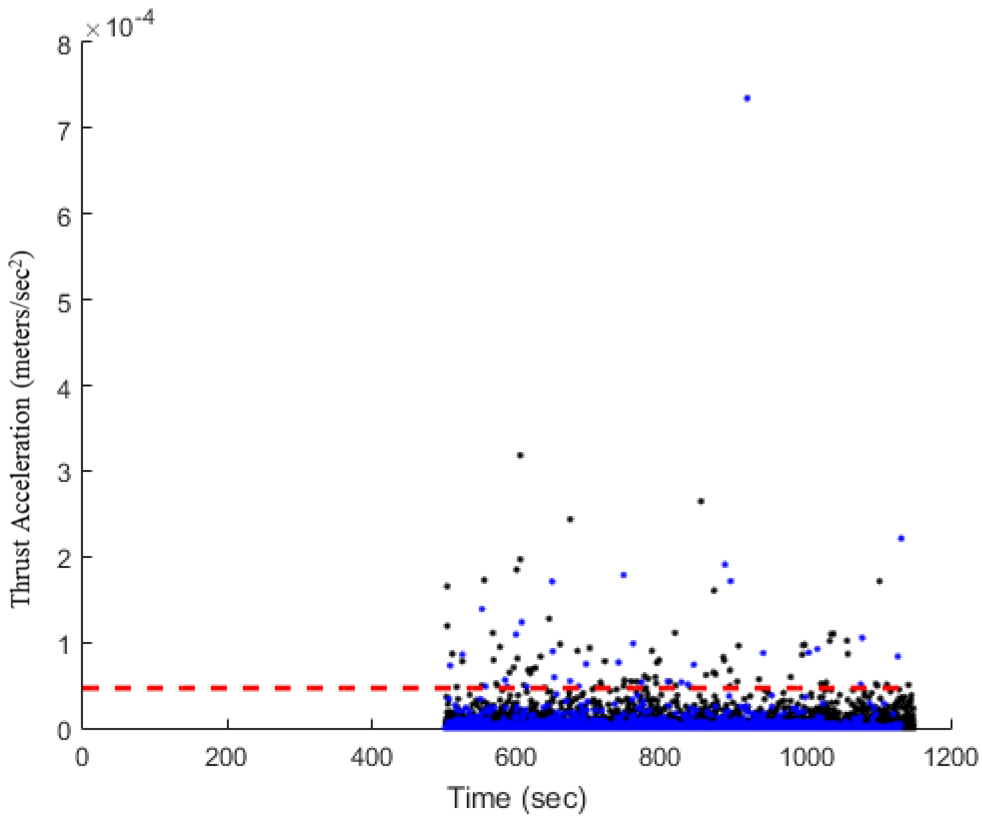

Figure **4-42:** Scenario 2 thrust magnitude estimates of the robust (black) and sub-optimal (blue) octree mapping

Thrust acceleration magnitude estimates in scenario 2 (Figure **4-42**) produced similar results between both octree mapping cases. Convergence issues were more prevalent in scenario 2 versus scenario 1 as the thrust magnitude estimation has a wider distribution below the true thrust magnitude. Error is roughly equivalent for both octree mapping cases with no inherent advantage for the robust mapping. As stated previously, both cases have more difficulty detecting the maneuver end thus both continued to estimate to the simulation end time during some runs.

4.3.4. Octree Target Parameters

As described in Section 3.18.3, the IMM is utilized in the VSD portion of the filter to leverage a range of density models to indirectly estimate the mass through applying a weight to the available densities and using the estimated target volume provided by the octree map (and initialized as a consider parameter). As the filter iterates through time, the weighting factors applied to each density is updated at the end of each filter loop. Table **3-3** contains the list of densities utilized in the IMM filter. These densities and the volume estimate contribute to the thrusting contribution to the orbital dynamics. By allowing the filter to converge to the mixture of densities that best predicts the state estimates that fit the observations, it provides the filter mass information of the object without directly estimating it. Based on the performance of each density model from the list in Table **3-3** in updating the state estimates, the corresponding weighting factors to the densities are updated. Over time, densities that do not contribute properly to the best combined state estimate are lowered in their weighting whereas densities that produce the best results are weighted higher. Figure **4-43** shows the *8* weighting factors corresponding to each density in Table **3-3** over the course of the Monte Carlo runs for both octree mapping cases. The filter consistently chose certain densities over others; the third density generally contributed more whereas the first density predominately contributed the least to the combined state estimates. There were several runs where the order (from highest to lowest weighting) changed between the top four contributing densities. However, the convergence for each density was very consistent.

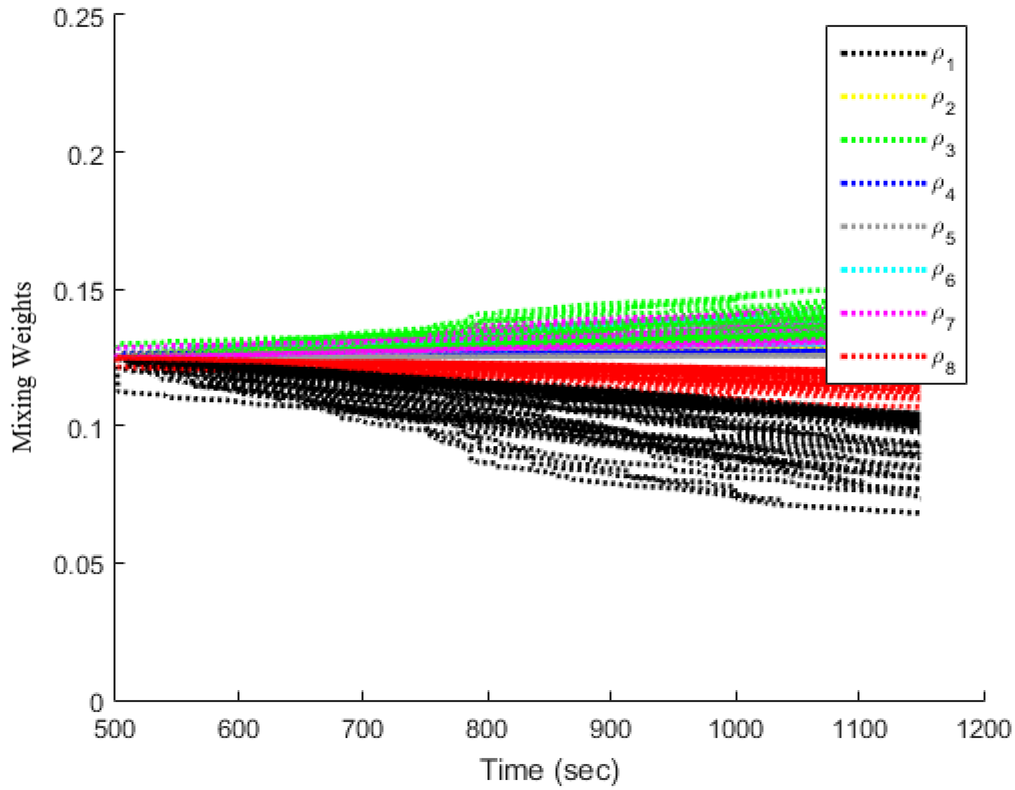

Figure **4-43:** Scenario 1 weighting factors of the IMM

In scenario 2, the weighting factors converged in a very similar manner to that of scenario 1 with less variance in how each individual weighting converged. Figure **4-44** shows that each weighting factor converges in a tighter grouping across the Monte Carlo runs. This is contributed to the higher uncertainty inherent in the larger thrust scenario as the state estimates possessed larger covariance due to the target moving at a faster rate.

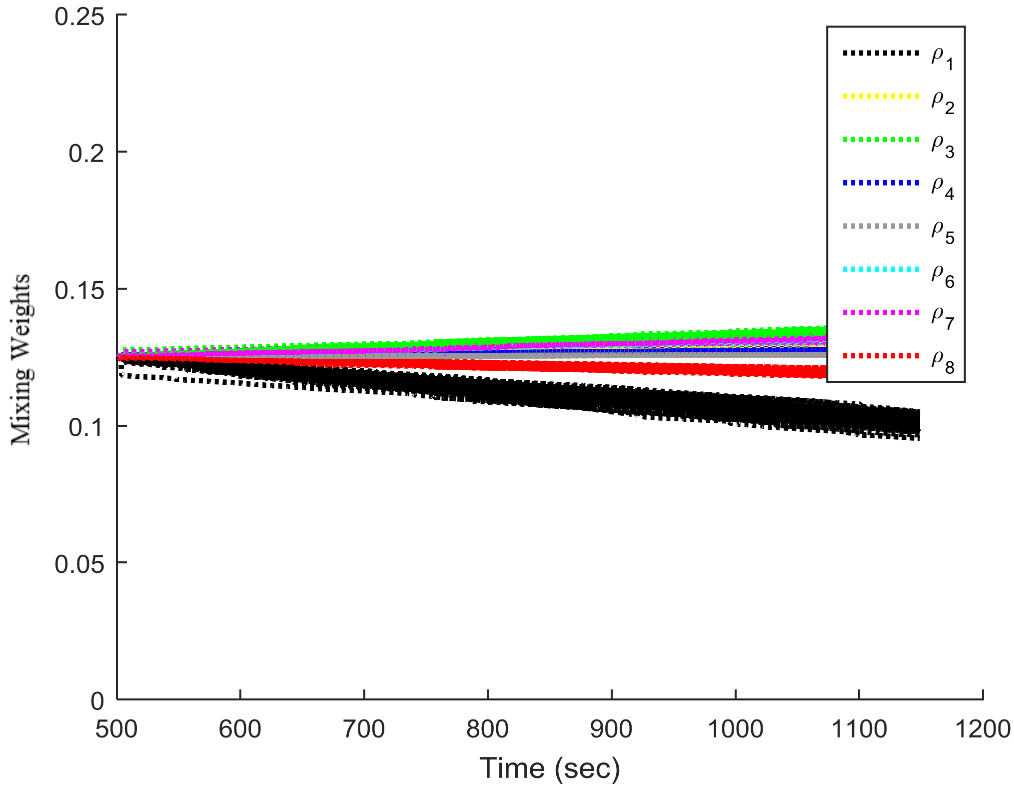

Figure **4-44:** Scenario 2 weighting factors of the IMM

Taking the densities, corresponding weighting factors, and target volume estimate, a coarse mass estimate can be extrapolated to determine the values of mass used by the filter. By multiplying the densities (ρ) with their corresponding weights and summing them, a combined density is retrieved to compute the mass (m) from the volume estimate (v).

$$\mathrm{m} = \sum_{\mathrm{i}=1}^{8} (\lambda_{\mathrm{i}} \rho_{\mathrm{i}}) \mathrm{v} \tag{4.7}$$

The resulting coarse mass estimates for both octree cases compared to the true mass for scenarios 1 and 2 are shown in Figures **4-45** and **4-46** respectively. As with the weighting factors, the mass estimation in scenario 2 has a lower variance than the mass estimates of scenario 1. Again, this is attributed to the higher uncertainty derived from the larger thrust magnitude.

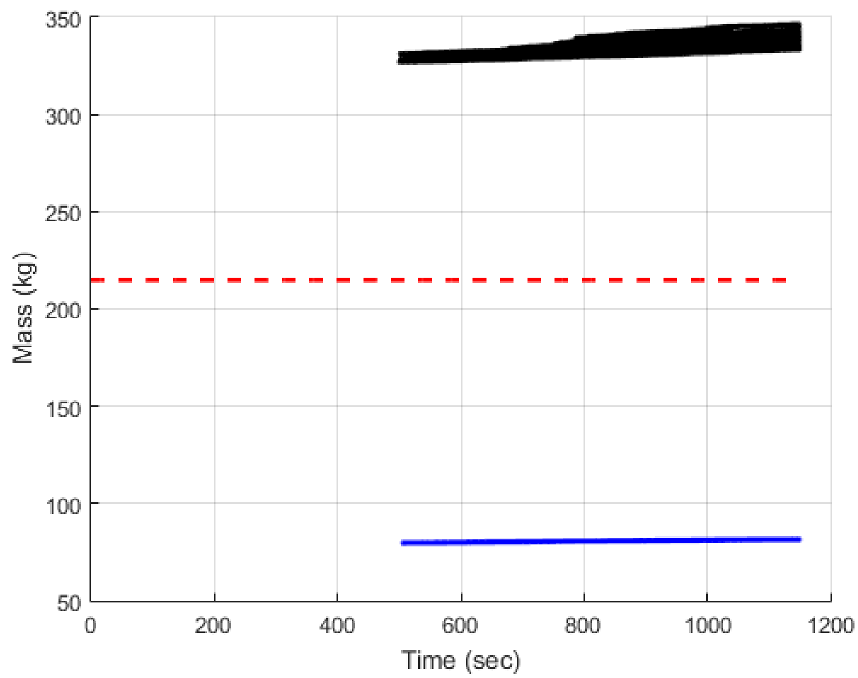

Figure **4-45:** Scenario 1 extrapolated mass estimate of the robust (black) and sub-optimal (blue) octree mapping

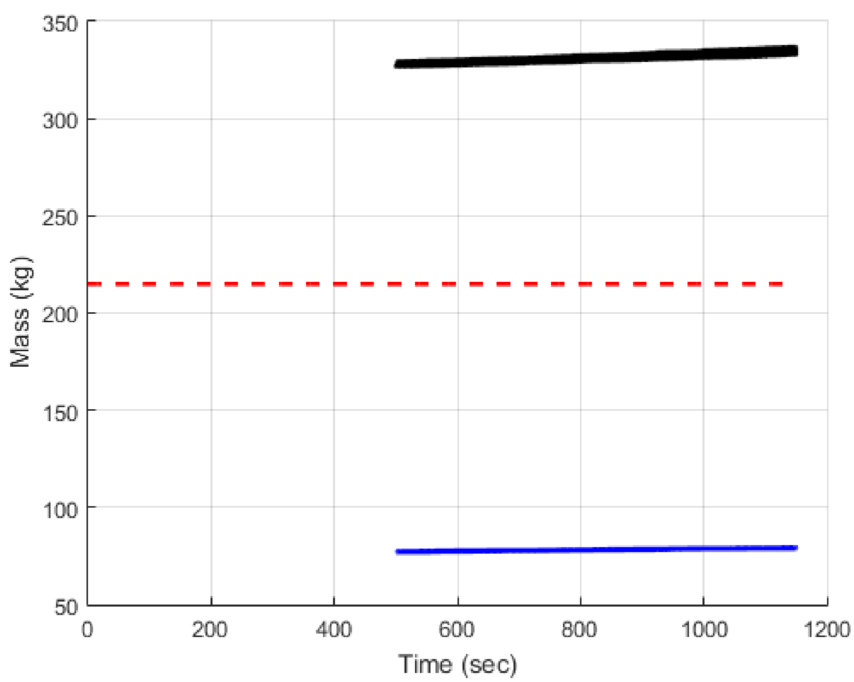

Figure **4-46:** Scenario 2 extrapolated mass estimate of the robust (black) and sub-optimal (blue) octree mapping

The true mass of the target is *215 kg* and the resulting mass estimates have over a *50 %* difference for the robust octree mapping cases and over a *60 %* difference for the sub-optimal cases. Table **4-6** summarizes the average mass estimation for each scenario and both octree mapping cases. Since estimating mass is not observable, attempting to do so could cause biased estimates or divergence in the estimation filter. By relying on an indirect approach to extrapolate mass information, accurate results are still achieved despite the large error in the mass used in the estimation filter.

Table **4-6**: Coarse mass estimation

	Scenario 1	**Percent Difference**	**Scenario 2**	**Percent Difference**
Robust octree mapping mass estimate (average)	333 *kg*	54.88 %	328 *kg*	52.56 %
Sub-optimal mapping mass estimate (average)	76 *kg*	64.65 %	76 *kg*	64.65 %

Though the coarse mass estimates used in the filter were incorrect, they were consistent in their error as the mass used in the dynamics over time did not change by much as shown in Figures **4-45** and **4-46**. Successive Monte Carlo runs with varying noise produced very steady coarse mass estimates that did not have much variance.

The geometric center of the target is obtained from the octree map and using the offset from geometric center (included in the state vector) and attitude of the target (from the octree map), the center of mass is extrapolated.

$$\mathrm{r}_{\mathrm{CoM}} = \mathrm{r}_{\mathrm{LVLH}} + \mathrm{R}_{\mathrm{G}}\mathrm{r}_{\mathrm{offset}} \tag{4.8}$$

Since the target object is not rotating, the attitude of the target is the geometric axes rather than the principal axes. Estimating the principal axes is unobservable unless the target is rotating. This means that an inherent bias is prevalent in the attitude as the principal axes cannot be extrapolated from the geometric axes. This bias is also inherent in the center of mass as it relates to the geometric axes R_G in Equation (4.8). Figure **4-47** shows the true attitude and center of mass compared to the geometric attitude and center of mass extrapolated from the robust octree map.

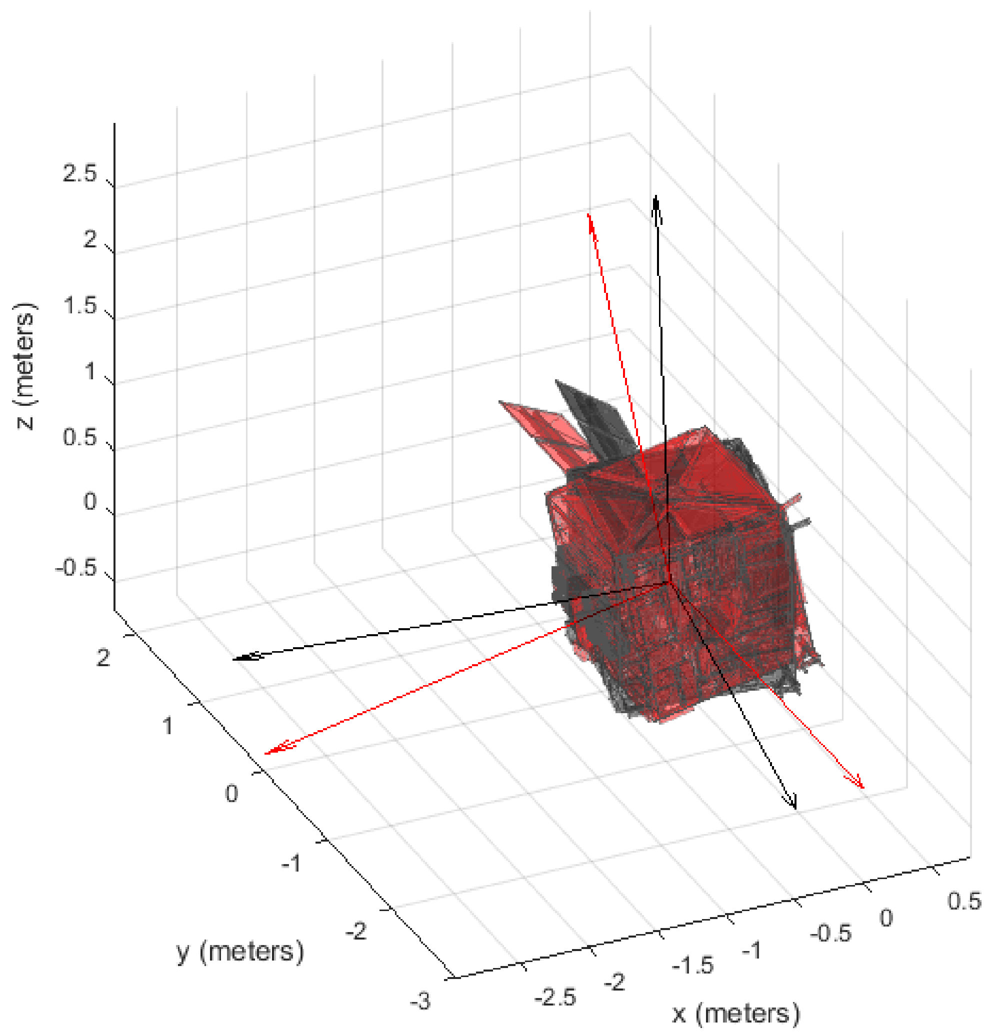

Figure **4-47**: True target attitude (red) versus geometric attitude (black) for robust octree target mapping

Despite the octree map having low uncertainty, a rotational and translational offset between the true axes and center of mass and the geometric axes and geometric center exists. In both scenarios, the geometric axes converged to roughly the same yaw, pitch, and roll (Table **4-7**). However, a noticeable difference exists between the octree mapping cases as the robust mapping has had a chance to observe most of the target's surfaces.

Table **4-7**: Center of mass in target body frame and attitude of target

	True target parameters	**Robust octree target mapping**	**Sub-optimal octree target mapping**
Yaw	180°	170.91°	169.46°
Pitch	11.42°	2.29°	-7.30°
Roll	0°	-1.24°	-0.94°
Center of mass (Target Body Frame)	[0.0050, -0.0131, 0.013] *m*	[0.016, -0.026, 0.022] *m*	[-0.026, -0.23, 0.11] *m*

Figure **4-48** visualizes the disparity in the attitude between the sub-optimal octree target mapping and the true target attitude. Though the map is continued to be built after the maneuver occurs, it uses the parameters initialized at maneuver detection to keep the consider filter consistent. The center of mass of the target is utilized in the guidance and control algorithm. While the inherent biases are an unfortunate product of the un-observability of the principal axes, the estimate is good enough for the controller to track the object.

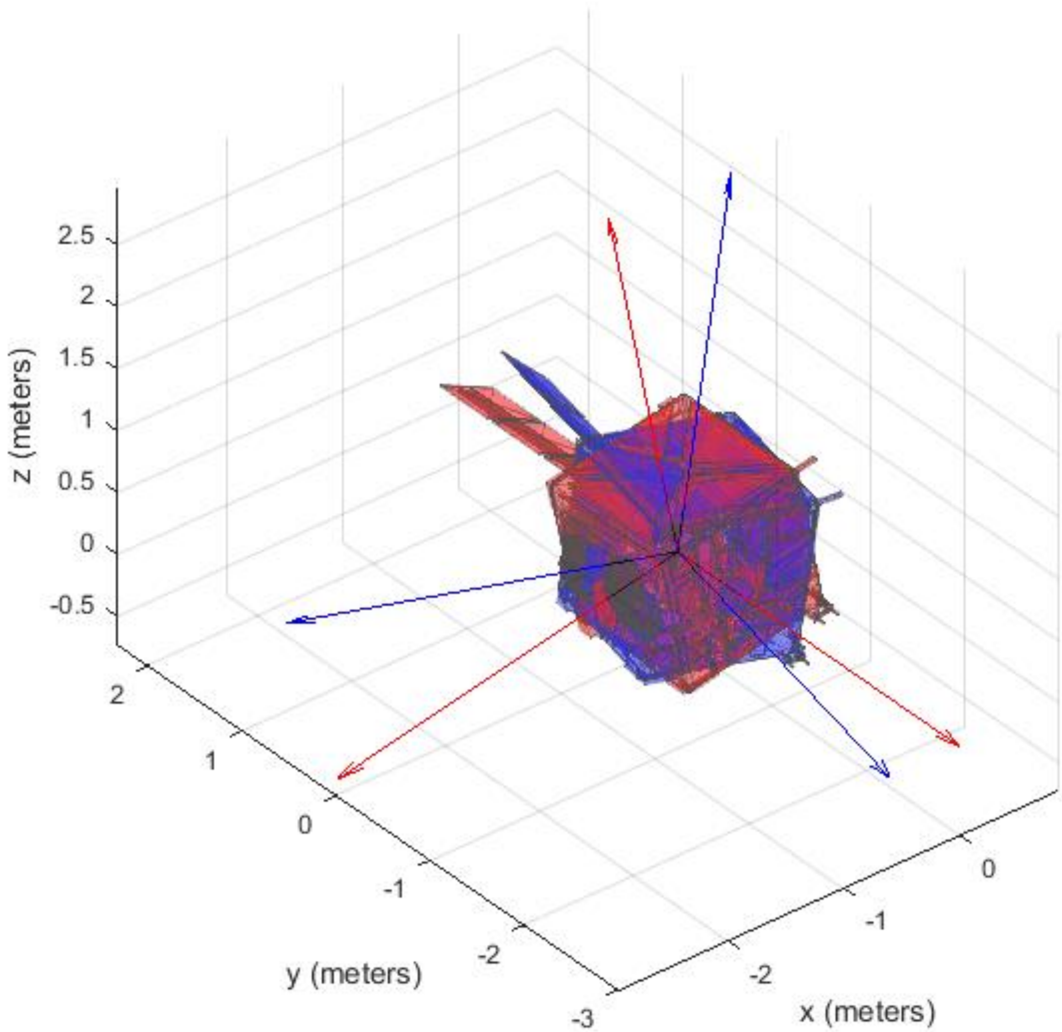

Figure **4-48**: True target attitude (red) versus geometric attitude (black) for sub-optimal octree target mapping

Chapter 5

Summary, Conclusions and Future Work

This dissertation sought to contribute toward the following areas in development of an algorithm capable of detecting an unknown maneuver from an uncooperative spacecraft using visual observations.

1) Develop a computationally efficient approach to detect and estimate a thrusting maneuver with no prior information available on the target.
2) Create an algorithm capable of extrapolating information from the target using visual observations to retrieve mass information in an online process.
3) Exploit information obtained about the target to improve the quality of future observations.

Detecting and estimating maneuvers whether intentional or uncontrolled are important for proximity operations missions. Development of robust and computationally efficient algorithms was developed in this dissertation toward that goal.

5.1. Summary

The work presented in this dissertation aimed to develop algorithms for the purposes of performing simultaneous localization and mapping on an uncooperative target conducting an unknown maneuver during a proximity operation. Previous work in this area is limited. Further constraints in consideration in this work are that the observer does not possess any information on the target and its initial relative position to the target. An algorithm was developed that estimates pose information and simultaneously builds an octree map of the target and extrapolates target volume, geometric center and axes to indirectly estimate the mass. These parameters are used in a consider variable state dimension (VSD) and interacting multiple model (IMM) filter which leverages bank of densities (and assuming equally distributed density) that comprise the models in the IMM to consider the uncertainty in the target volume on a detected maneuver. Estimated volume from the octree map and its associated uncertainty initialize the consider parameters at the start of the model switch from the quiescent model (initial state vector) to the thrusting model (initial states plus concatenated thrust states). As the uncertainty in the octree map decreases, the octree map is utilized as a pruning mechanism for inconsistent observations before they can enter the estimation filter.

Results presented in this dissertation show the developed algorithm's capability in handling an unknown continuous thrusting maneuver of very low magnitude for an object of previously unknown characteristics. Estimates converged over the course of several Monte Carlo simulations with the maneuver detection at the start and end of the maneuver being detected. To analyze the octree map's contributions to the algorithm's performance, two cases were explored: one in which the octree map had low uncertainty before a maneuver was detected and the second in which the octree mapping had higher uncertainty. While the robust octree mapping performed very well, the sub-optimal mapping had larger errors in the state estimates and maneuver detection. However, the filter on average was able to cope despite the deficiency in the mapping.

5.2. Conclusions

The developed algorithms fulfill the requirements set forth in the desired goals of this dissertation. Implementation of the VSD maneuver estimation approach provides a low-cost method of estimation. Keeping the number of models within the IMM low to kept the computational cost in check while ensuring enough models were present for robustness and filter performance. Tying the octree mapping to the consider parameters provided the filter with a computationally efficient approach to utilize the uncertainty in the target volume to estimate the thrust. Accuracy was sacrificed for speed in keeping the octree bin size larger. Despite an estimated target volume that was *22-30 %* different, the estimation filter performed well. Estimations converged and the maneuver initiation and end were detected with accurate thrust acceleration estimation.

The octree mapping providing a volume estimate for the consider filter enabled an online indirect mass estimation approach to extrapolate target information from the octree map and using it for thrust acceleration estimation. Volume estimates had noticeable error causing the indirectly obtained mass to possess upwards of *50 %* difference from the true volume as specified in Table **4-5**. Estimation of volume and by extension mass is an inherently challenging problem due to observability of these parameters being severely limited. As such, it was known from the outset that obtaining accurate volume and mass estimates would not be realistic. Instead, it was desired to achieve same order of magnitude estimates that while coarse, could be refined in the estimation filter to provide enough information for the filter to converge. Results showed that despite the inaccurate volume estimate (and mass by extension), the filter reliably converged and performed well in the maneuver detection and estimation. Clearly, the octree mapping provided a benefit as its uncertainty decreased as shown in the results of the robust octree mapping versus the sub-optimal case. The estimated volume (while course) from the robust octree mapping case consistently provided better estimation and maneuver detection over the sub-optimal mapping.

As the octree map's associated uncertainty became low, the map was utilized in pruning bad measurements from entering the filter. Implementation of this provided a low cost approach to improve filter performance by eliminating erroneous measurements. Comparing the results of the octree mapping cases shows an effect on having the pruning of bad measurements available in the robust octree mapping case versus the sub-optimal case in which the mapping had high uncertainty and thus no measurement pruning.

5.3. Future Work

There are several avenues of improvements that can be implemented from this work. Firstly, modifying the algorithm to handle both an unknown translation and rotation of an uncooperative target is a natural next step. This is not trivial as a moving observer satellite inspecting a rotating and translating target is a very challenging problem. Even assuming both motions are decoupled, one of the difficulties present is that a perceived rotation could cause a false maneuver to be detected. This behavior was seen in preliminary work into implementation of a rotating target. Most previous work in the literature examines the rotation problem so solving the extended rotation and translation problem is the natural progression.

Further work can also be done in the creation of the octree map itself. Developing a more efficient and accurate methodology for creating a map to extrapolate target characteristics would be greatly beneficial toward robustness especially if the target is rotating. More work related to the octree mapping would be to explore further approaches to improve incoming measurements. In this work, a simple pruning method was implemented only checking for features contained within the bins. Implementation of a nearest neighbor approach or some other optimized searching algorithm may enhance filter performance. Improvement of the mapping may also lead to better volume estimates and indirectly better mass estimates.

Another direction for future research would be in development of the control to rendezvous with the target. If the target is performing a maneuver, developing the control to predict its future translational states from the predicted thrust estimates to rendezvous with the target would be an interesting extension of this work.

Further validation would be required to propose this work as a flight experiment. In orbit, further situations may occur that could affect performance and would need to be tested. Such things include images containing stars and/or the Moon and Earth. Images may also contain hot/dead pixels and electric noise. Furthermore, external forces like drag, solar radiation pressure, *n*-body effects (gravitational sources from the Moon and Sun), sloshing, and gravity-gradient torques (both acting on the target) may impede maneuver detection and make thrust estimation more difficult. Finally, the estimation of a rotating target would need to be implemented as this dissertation currently makes a non-rotating assumption. In reality, any of the mentioned external forces as well as the thrusting maneuver may induce a rotating torque upon the object. A rotating object would trigger the maneuver detection in the current framework so implementation of this capability is vital. Testing on hardware and processor-in-the-loop systems would further be necessary before proposing a flight experiment.

REFERENCES

[1] Abidi, M. A., Green, W. L., Chandra, T., and Spears, J., "Multisensor Robotic System For Autonomous Space Maintenance And Repair," *Space Station Automation IV*, vol. 1006, 1988, p. 104.

[2] Zhao, Z., Zhang, J., Li, H. yang, and Zhou, J. yong, "LEO cooperative multi-spacecraft refueling mission optimization considering J 2 perturbation and target's surplus propellant constraint," *Advances in Space Research*, vol. 59, 2017, pp. 252–262.

[3] Zhu, X., Zhang, C., Sun, R., Chen, J., and Wan, X., "Orbit determination for fuel station in multiple SSO spacecraft refueling considering the J2 perturbation," *Aerospace Science and Technology*, vol. 105, 2020, p. 105994.

[4] Boning, P., and Dubowsky, S., "Coordinated control of space robot teams for the on-Orbit construction of large flexible space structures," *Advanced Robotics*, vol. 24, 2010, pp. 303–323.

[5] Hakima, H., Bazzocchi, M. C. F., and Emami, M. R., "A deorbiter CubeSat for active orbital debris removal," *Advances in Space Research*, vol. 61, 2018, pp. 2377–2392.

[6] Konin, V., and Shyshkov, F., "Autonomous navigation of service spacecrafts on geostationary orbit using GNSS signals," *Radioelectronics and Communications Systems*, vol. 59, 2016, pp. 562–566.

[7] Mark, C. P., and Kamath, S., "Review of Active Space Debris Removal Methods," *Space Policy*, vol. 47, 2019, pp. 194–206.

[8] Augenstein, S., "Monocular Pose and Shape Estimation of Moving Targets, for Autonomous Rendezvous and Docking," Aeronautics and Astronautics, PhD, Stanford University, 2011.

[9] Tweedle, B. E., "Computer Vision-Based Localization and Mapping of an Unknown,

Uncooperative and Spinning Target for Spacecraft Proximity Operations," Aeronautics and Astronautics, PhD, MIT, 2013.

10 Tweedle, B. E., Saenz-Otero, A., Leonard, J. J., and Miller, D. W., "Factor Graph Modeling of Rigid-body Dynamics for Localization, Mapping, and Parameter Estimation of a Spinning Object in Space," *Journal of Field Robotics*, vol. 32, 2015, pp. 897–933.

11 Sharma, S., and D'Amico, S., "Reduced-dynamics pose estimation for non-cooperative spacecraft rendezvous using monocular vision," *38th Advances in the Astronautical Sciences Guidance and Control Conference*, 2017, pp. 361–385.

12 Lavagna, M., Pesce, V., and Bevilacqua, R., "Uncooperative objects pose motion and inertia tensor estimation via stereovision," *Advanced Space Technologies for Robotics and Automation*, 2015.

13 Lichter, M. D., "Shape , Motion , and Inertial Parameter Estimation of Space Objects using Teams of Cooperative Vision Sensors," Massachusetts Institute of Technology, Mechanical Engineering, PhD, 2005.

14 Lichter, M. D., Dubowsky, S., Ueno, H., and Mitani, S., "Shape, Motion, and Parameter Estimation of Flexible Space Structures using Laser Rangefinders," *Robotics: Science and Systems*, Cambridge, MA: 2005.

15 Pesce, V., Lavagna, M., and Bevilacqua, R., "Stereovision-based pose and inertia estimation of unknown and uncooperative space objects," *Advances in Space Research*, vol. 59, 2016, pp. 236–251.

16 Yu, F., He, Z., Qiao, B., and Yu, X., "Stereo-vision-based relative pose estimation for the rendezvous and docking of noncooperative satellites," *Mathematical Problems in Engineering*, vol. 2014, 2014, pp. 1–12.

17 Wang, X., Wang, Z., and Zhang, Y., "Stereovision-based relative states and inertia parameter estimation of noncooperative spacecraft," *Proceedings of the Institution of*

Mechanical Engineers, Part G: Journal of Aerospace Engineering, vol. 233, 2019, pp. 2489–2502.

18 Bailey, T., and Durrant-Whyte, H., "Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms," *Robotics & Automation Magazine, IEEE*, vol. 13, 2006, pp. 99–110.

19 Bailey, T., and Durrant-Whyte, H., "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics and Automation Magazine*, vol. 13, 2006, pp. 108–117.

20 Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J. J., Leonard}, J. J., Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J., "Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, 2016, pp. 1309–1332.

21 Younes, G., Asmar, D., Shammas, E., and Zelek, J., "Keyframe-based monocular SLAM: design, survey, and future directions," *Robotics and Autonomous Systems*, vol. 98, 2017, pp. 67–88.

22 Leishman, R. C., "A Vision-Based Relative Navigation Approach for Autonomous Multirotor Aircraft," Mechanical Engineering, PhD, Brigham Young University, 2013.

23 Leishman, R. C., McLain, T. W., and Beard, R. W., "Relative navigation approach for vision-based aerial GPS-denied navigation," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 74, 2014, pp. 97–111.

24 Dor, M., and Tsiotras, P., "ORB-SLAM Applied to Spacecraft Non-Cooperative Rendezvous," *AIAA SciTech Forum*, 2018, pp. 1–17, doi: 10.2514/6.2018-1963.

25 Strasdat, H., Montiel, J. M. M., and Davison, A. J., "Visual SLAM: Why filter?," *Image and Vision Computing*, vol. 30, 2012, pp. 65–77.

26 Zhou, W., Zhao, C., and Guo, J., "The study of improving Kalman filters family for

nonlinear SLAM," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 56, 2009, pp. 543–564.

27 Kottas, D. G., Hesch, J. A., Bowman, S. L., and Roumeliotis, S. I., "On the Consistency of Vision-Aided Inertial Navigation," *Proc. Int. Symp. Exp. Robot*, 2012, pp. 303–317.

28 Yadkuri, F. F., and Khosrowjerdi, M. J., "Methods for Improving the Linearization Problem of Extended Kalman Filter," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 78, 2015, pp. 485–497.

29 He, B., Liu, Y., Dong, D., Shen, Y., Yan, T., and Nian, R., "Simultaneous localization and mapping with iterative sparse extended information filter for autonomous vehicles," *Sensors (Switzerland)*, vol. 15, 2015, pp. 19852–19879.

30 Ila, V., Porta, J. M., and Andrade-Cetto, J., "Information-based compact pose SLAM," *IEEE Transactions on Robotics*, vol. 26, 2010, pp. 78–93.

31 Thrun, S., Liu, Y., Koller, D., Ng, A. Y., Ghahramani, Z., and Durrant-Whyte, H., "Simultaneous localization and mapping with sparse extended information filters," *International Journal of Robotics Research*, vol. 23, 2004, pp. 693–716.

32 Demim, F., Nemra, A., and Louadj, K., "Robust SVSF-SLAM for Unmanned Vehicle in Unknown Environment," *IFAC-PapersOnLine*, vol. 49, 2016, pp. 386–394.

33 Fethi, D., Nemra, A., Louadj, K., and Hamerlain, M., "Simultaneous localization, mapping, and path planning for unmanned vehicle using optimal control," *Advances in Mechanical Engineering*, vol. 10, 2018, pp. 1–25.

34 Sigges, F., Rauterberg, C., Baum, M., and Hanebeck, U. D., "An Ensemble Kalman Filter for Feature-Based SLAM with Unknown Associations," *2018 21st International Conference on Information Fusion, FUSION 2018*, 2018, pp. 346–352.

35 Goff, G. M., Black, J. T., and Beck, J. A., "Orbit Estimation of a Continuously Thrusting Spacecraft Using Variable Dimension Filters," *Journal of Guidance, Control, and*

Dynamics, vol. 38, 2015, pp. 2407–2420.

36 Bar-Shalom, Y., and Birmiwal, K., "Variable Dimension Filter for Maneuvering Target Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-18, 1982, pp. 621–629.

37 Bekir, E., "Adaptive Kalman filter for tracking maneuvering targets," *Journal of Guidance, Control, and Dynamics*, vol. 6, 2008, pp. 414–416.

38 Cloutier, J. R., Lin, C.-F., and Yang, C., "Maneuvering target tracking via smoothing and filtering through measurement concatenation," *Journal of Guidance, Control, and Dynamics*, vol. 16, 1993, pp. 377–384.

39 Punithakumar, K., Kirubarajan, T., and Sinha, A., "Multiple-model probability hypothesis density filter for tracking maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, 2008, pp. 87–98.

40 Salam, A. O. A., "Adaptive tracking of maneuvering targets using two-stage Kalman filter," *2015 IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2015*, 2015, pp. 74–78.

41 Kelecy, T., and Jah, M., "Detection and orbit determination of a satellite executing low thrust maneuvers," *Acta Astronautica*, vol. 66, 2010, pp. 798–809.

42 McReynolds, S. R., "Fixed Interval Smoothing : Revisited," *Journal of Guidance, Control, and Dynamics*, vol. 13, 1990, pp. 913–921.

43 Simon, D., *Optimal State Estimation*, Hoboken, NJ: John Wiley Sons, Inc., 2006.

44 Szeliski, R., *Computer Vision: Algorithms and Applications*, Springer-Verlag London Limited, 2011.

45 Wong, X. I., and Majji, M., "Extended Kalman Filter for Stereo Vision-Based Localization and Mapping Applications," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, 2018, pp. 1–16.

[46] Lucas, B., and Kanade, T., "An iterative image registration technique with an application to stereo vision," *In Proceedings of the International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

[47] Baker, S., and Matthews, I., "Lucas-Kanade 20 Years On: A Unifying Framework: Part 1," *International Journal of Computer Vision*, vol. 56, 2004, pp. 221–255.

[48] Sullivan, J., Grimberg, S., and D'Amico, S., "Comprehensive Survey and Assessment of Spacecraft Relative Motion Dynamics Models," *Journal of Guidance, Control, and Dynamics*, vol. 40, 2017, pp. 1837–1859.

[49] Xu, G., and Wang, D., "Nonlinear Dynamic Equations of Satellite Relative Motion Around an Oblate Earth," *Journal of Guidance, Control, and Dynamics*, vol. 31, 2008, pp. 1521–1524.

[50] Wu, B., Wang, D., Xu, G., and Yue, H., "Nonlinear optimization of low-thrust trajectory for satellite formation: Legendre pseudospectral approach," *Advances in the Astronautical Sciences*, vol. 130 PART 1, 2008, pp. 671–690.

[51] Xu, G., Wang, D., Eng, K. P., and Wu, B., "Periodic and quasi-periodic satellite relative orbits at critical inclination," *IEEE Aerospace Conference Proceedings*, 2009, pp. 1–11.

[52] Vallado, D., *Fundamentals of Astrodynamics and Applications*, Hawthorne, CA: Microcosm Inc., 2013.

[53] Kuipers, J. B., *Quaternions and Rotation Sequences*, Princeton, NJ: Princeton University Press, 1999.

[54] Crassidis, J., and Markley, F., "Attitude Estimation Using Modified Rodrigues Parameters," *Flight Mechanics/Estimation Theory Symposium 1996*, Greenbelt, MD: 1996, pp. 71–83.

[55] Markley, F. L., and Crassidis, J. L., *Fundamentals of Spacecraft Attitude Determination and Control*, New York City: Springer Science+Business Media, 2014.

[56] Mortari, D., Markley, F. L., and Junkins, J. L., "Optimal linear attitude estimator," *Advances in the Astronautical Sciences*, vol. 105 I, 2000, pp. 465–477.

[57] Wong, X. I., Lee, T., Singla, P., and Majji, M., "Optimal Linear Attitude Estimator for Alignment of Point Clouds," *{CVPR} Workshops*, 2018, pp. 1496–1504.

[58] Vespa, E., Nikolov, N., Grimm, M., Nardi, L., Kelly, P. H. J., and Leutenegger, S., "Efficient Octree-Based Volumetric SLAM Supporting Signed-Distance and Occupancy Mapping," *IEEE Robotics and Automation Letters*, vol. 3, 2017, pp. 1–8.

[59] Laine, S., and Karras, T., "Efficient Sparse Voxel Octrees – Analysis , Extensions , and Implementation," *IEEE Transactions on Visualization and Computer Graphics*, 2010, pp. 1–30.

[60] Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W., "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, 2013, pp. 189–206.

[61] Smith, R. C., and Cheeseman, P., "On the Representation and Estimation of Spatial Uncertainty," *The International Journal of Robotics Research*, vol. 5, 1986, pp. 56–68.

[62] Gilsinn, D. E., Cheok, G. S., and Lytle, A. M., "Pose of I-Beams for Construction Site Automation," *Proceedings of the 21st International Symposium on Automation and Robotics in Construction*, 2017, pp. 1–6.

[63] Ding, Y., Xu, Z., Zhang, Y., and Sun, K., "Fast lane detection based on bird's eye view and improved random sample consensus algorithm," *Multimedia Tools and Applications*, vol. 76, 2017, pp. 22979–22998.

[64] Kaspi, O., Yosipof, A., and Senderowitz, H., "RANdom SAmple Consensus (RANSAC) algorithm for material-informatics: application to photovoltaic solar cells," *Journal of Cheminformatics*, vol. 9, 2017, pp. 1–15.

[65] Teoh, S. T., Kitamura, M., Nakayama, Y., Putri, S., Mukai, Y., and Fukusaki, E.,

"Random sample consensus combined with partial least squares regression (RANSAC-PLS) for microbial metabolomics data mining and phenotype improvement," *Journal of Bioscience and Bioengineering*, vol. 122, 2016, pp. 168–175.

66 Tong, X., Ye, Z., Xu, Y., Liu, S., Li, L., Xie, H., and Li, T., "A Novel Subpixel Phase Correlation Method Using Singular Value Decomposition and Unified Random Sample Consensus," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, 2015, pp. 4143–4156.

67 Civera, J., Grasa, O. G., Davison, A. J., and Montiel, J. M. M., "1-Point RANSAC for EKF filtering. Application to Real-Time Structure from Motion and Visual Odometry," *Journal of Field Robotics*, vol. 27, 2010, pp. 609–631.

68 Kee, C. Y., and Wang, C., "Efficient GPU Implementation of the High-Frequency SBR-PO Method," *IEEE Antennas and Wireless Propagation Letters*, vol. 12, 2013, pp. 941–944.

69 Phong, B. T., "Illumination for Computer-Generated Images," Computer Science, MS, Utah University, 1973.

70 Sinha, S., *Phong Model*, Boston, MA: Springer US, 2016.

71 Crassidis, J. L., and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, Boca Raton, FL: Taylor & Francis Group, LLC, 2012.

72 Singla, P., "Multi-Resolution Methods for High Fidelity Modeling and Control Allocation in Large-Scale Dynamical Systems," Aerospace Engineering, PhD, Texas A&M University, 2006.

73 Schmidt, S. F., "Application of State-Space Methods to Navigation Problems," *Advances in Control Systems*, vol. 3, 1966, pp. 293–340.

74 Woodbury, D., and Junkins, J., "On the Consider Kalman Filter," *AIAA Guidance, Navigation, and Control Conference*, pp. 1-28, 2010, doi: 10.2514/6.2010-7752.

[75] Lou, T., Fu, H., Zhang, Y., and Wang, Z., "Consider unobservable uncertain parameters using radio beacon navigation during Mars entry," *Advances in Space Research*, vol. 55, 2015, pp. 1038–1050.

[76] Zanetti, R., and D'Souza, C., "Recursive Implementations of the Schmidt-Kalman 'Consider' Filter," *The Journal of the Astronautical Sciences*, vol. 60, 2015, pp. 672–685.

[77] Brink, K. M., "Partial-Update Schmidt–Kalman Filter," *Journal of Guidance, Control, and Dynamics*, vol. 40, 2017, pp. 2214–2228.

[78] Brink, K. M., "Unscented Partial-Update Schmidt–Kalman Filter," *Journal of Guidance, Control, and Dynamics*, vol. 41, 2018, pp. 929–935.

[79] Stauch, J., and Jah, M., "Unscented Schmidt–Kalman Filter Algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 38, 2014, pp. 117–123.

[80] McCabe, J. S., and DeMars, K. J., "Considering Uncertain Parameters in Non-Gaussian Estimation for Single-Target and Multitarget Tracking," *Journal of Guidance, Control, and Dynamics*, vol. 40, 2017, pp. 2138–2150.

[81] Geeraert, J. L., and McMahon, J. W., "Square-Root Unscented Schmidt–Kalman Filter," *Journal of Guidance, Control, and Dynamics*, vol. 41, 2018, pp. 280–287.

[82] Lou, T.-S., Wang, Z.-H., Xiao, M.-L., and Fu, H.-M., "Multiple Adaptive Fading Schmidt-Kalman Filter for Unknown Bias," *Mathematical Problems in Engineering*, vol. 2014, 2014, pp. 1–8.

[83] Crassidis, J. L., and Junkins, J. L., *Optimal Estimation of Dynamic Systems*, Boca Raton, FL: Taylor & Francis Group, LLC, 2012.

[84] Tapley, B. D., Schutz, B. E., and Born, G. H., *Statistical Orbit Determination*, Elsevier Academic Press, 2004.

[85] Scarcella, P., Johnson, K., and Hess, J., "Consider Filtering Applied to Maneuver Detection for Relative Orbit Determination," *2019 Advances in the Astronautical*

Sciences, 2019, pp. 2825–2842.

86 Xu, Q., Li, X., and Chan, C. Y., "A cost-effective vehicle localization solution using an interacting multiple model—unscented kalman filters (IMM-UKF) Algorithm and grey neural network," *Sensors (Switzerland)*, vol. 17, 2017.

87 Williams, J. L., "Gaussian Mixture Reduction for Tracking Multiple Maneuvering Targets in Clutter," U.S. Air Force Inst. of Technology, 2003.

88 Christian, J. A., and Glenn Lightsey, E., "An on-board image processing algorithm for a spacecraft optical navigation sensor system," *AIAA SPACE Conference and Exposition 2010*, pp 1-28, doi: 10.2514/6.2010-8920.

89 Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O., "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, 2007, pp. 1052–1067.

90 Kostavelis, I., Boukas, E., Nalpantidis, L., and Gasteratos, A., "Stereo-Based Visual Odometry for Autonomous Robot Navigation," *International Journal of Advanced Robotic Systems*, vol. 13, 2016.

91 Lemaire, T., Berger, C., Jung, I. K., and Lacroix, S., "Vision-based SLAM: Stereo and monocular approaches," *International Journal of Computer Vision*, vol. 74, 2007, pp. 343–364.

92 Mooney, J. G., and Johnson, E. N., "A Comparison of Automatic Nap-of-the-earth Guidance Strategies for Helicopters," *Journal of Field Robotics*, vol. 33, 2014, pp. 1–17.

93 Moreno, F. A., Blanco, J. L., and Gonzalez, J., "Stereo vision specific models for particle filter-based SLAM," *Robotics and Autonomous Systems*, vol. 57, 2009, pp. 955–970.

94 Mur-Artal, R., and Tardos, J. D., "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, 2017, pp. 1255–1262.

[95] Paz, L. M., Pini, P., and Tard, J. D., “Large-Scale 6-DOF SLAM With Stereo-in-Hand,” vol. 24, 2008, pp. 946–957.

[96] Sim, R., Elinas, P., Griffin, M., and Little, J. J., “Vision-based SLAM using the Rao-Blackwellised particle filter,” *IJCAI 2005 Workshop on Reasoning with Uncertainty in Robotics, RUR 2005*, 2005, pp. 9–16.

[97] Sim, R., Elinas, P., and Little, J. J., “A study of the Rao-Blackwellised particle filter for efficient and accurate vision-based SLAM,” *International Journal of Computer Vision*, vol. 74, 2007, pp. 303–318.

[98] Sidi, M., *Spacecraft Dynamics & Control: A Practical Engineering Approach*, Cambridge University Press, 1997.

[99] Lubey, D. P., Scheeres, D. J., "Supplementing state and dynamics estimation with information from optimal control policies," *17th International Conference on Information Fusion (FUSION),* Salamanca, 2014, pp. 1–7.

[100] Lubey, D. P., Scheeres, D. J., Erwin, R S., “Maneuver Detection and Reconstruction of Stationkeeping Spacecraft at GEO using the Optimal Control-Based Estimator”, *IFAC-PapersOnLine*, Volume 48, Issue 9, 2015, Pages 216–221, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2015.08.086.

VITA

Peter Scarcella

Peter Craig Scarcella was born in Perth Amboy, New Jersey in 1988. After graduating from North Plainfield High School in 2006, he attended Raritan Valley Community College under the New Jersey Student Tuition Assistance Reward Scholarship (NJ STARS) obtaining a A.S. in Science Engineering in 2008. Upon completion of his associates, he transferred to Rutgers University in the Department of Mechanical and Aerospace Engineering where he obtained his B.S. in 2011. During the next three years, Peter had two engineering jobs before being accepted to the graduate program at Penn State University in the Aerospace Engineering Department under the tutelage of Dr. David Spencer.

In 2016, Peter completed his thesis titled *Parametric Trade Study of Multiple Liberation Point Orbits in the Circular Restricted Four-Body Problem*. After obtaining his M.S., he transitioned to the Ph.D. program continuing his education and research at Penn State. During the summers of 2018 and 2019, Peter had internships at the Air Force Institute of Technology (AFIT) in Dayton, Ohio under the advisement of Lt. Col Kirk Johnson culminating in a conference paper at the 2019 AAS/AIAA Astrodynamics Specialist Conference. In February 2020, Peter accepted a position as Systems Engineer at Applied Technology Associates (ATA) in Albuquerque, NM doing technical work for the Air Force Research Laboratory (AFRL).

Peter's research interests include rendezvous proximity operations (RPO), relative navigation, maneuver detection, pose and shape estimation, estimation filters, and computer vision. He will continue his work at ATA until being hired directly by AFRL upon completion of his Ph.D. requirements.