Architecture and Algorithms for Space-Based Global Wildlife Tracking

Fausto Vega

CMU-RI-TR-23-71

September 1, 2023

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Zachary Manchester, *chair*
Michael Kaess
Kevin Tracy

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my family.

Abstract

Accurate satellite-based positioning has revolutionized several industries over the past two decades from agriculture to transportation. However, conventional GNSS receivers consume significant amounts of energy and are too large for many applications, including wildlife-tracking which is critical for conservation efforts and improving our understanding of the global climate. To address this capability gap, we propose a new positioning system to minimize the mass, power, and size of the terrestrial tracking device. We analyze, through extensive modeling and simulation, a mission concept that relies on space-based receivers hosted on a constellation of small satellites in low-Earth orbit (LEO) that detect and localize signals from very small transmitter tags. We compare a variety of positioning techniques, including both Doppler and time-of-arrival methods, and evaluate the achievable position accuracy across a wide range of design parameters. Our model also accounts for errors in satellite orbital state knowledge, clock offsets, frequency measurement errors, and ionospheric effects. This thesis will present the results of our extensive trade study along with the orbit determination methods used to obtain the satellite state uncertainty.

Acknowledgments

First, I would like to thank my advisor Zac Manchester for the guidance and mentorship throughout this work. His passion for research, valuable insights, and constant support have allowed me to learn immensely and grow as a researcher. I am grateful for the opportunities that have arose from working with Zac, and appreciate his patience and all the tools and resources he's provided me to explore the field of space robotics. I also appreciate the support from the Rachel Burcin and John Dolan with the opportunity to participate in the Robotics Institute Summer Scholars (RISS) during my undergraduate as I was introduced to CMU and Zac through this program.
I would also like to thank the members of the Robotic Exploration Laboratory especially Kevin Tracy and Jacob Willis for the valuable research discussions and help throughout many of the questions I had.

Finally, I would like to thank my parents, brothers, and grandparents for all the love and support as they are my motivation for being here today.

Funding

This work was supported by the GEM Fellowship, NASA, and the Bureau of Ocean Energy Management (BOEM).

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

The development of satellite-based tracking began with TRANSIT, a Doppler-based satellite navigation system that calculated the position of a terrestrial receiver. This system provided infrequent position updates for Polaris ballistic missile submarines using the Doppler shift, the change in frequency a signal experiences once a transmitter is moving relative to a receiver [7]. Then, the need for continuous tracking led to the development of global positioning system (GPS) which uses a constellation of 24 satellites in medium earth orbit to provide positioning information globally [25]. GPS determines the pseudorange between the receiver and each satellite in sight using time of arrival measurements. Both Transit and GPS satellites transmit signals to localize the terrestrial receiver, however, having the receivers onboard satellites and the transmitter emit signals from Earth is another method of tracking a mobile asset. COSPAS-SARSAT, an international satellite system designed for search and rescue operations, utilizes this inverse method of tracking to locate emergency locator transmitters (ELT's) [34]. ELT's are onboard aircraft and broadcast distress signals at 406 MHz automatically when submerged in water or experience a high impact. These signals are then detected by non geostationary satellites and the aircraft position is located using trilateration [23]. Over the years, satellite-based tracking has been applied to a spectrum of domains from search and rescue to military, however, most receivers are too large for many applications including wildlife tracking. Wildlife tracking is essential for our understanding of the global climate as species migration behavior is an important indicator of climate change [19]. Existing methods for

wildlife tracking include ARGOS and GPS. ARGOS exploits the frequency shift in the transmitter signal measured by the receiver onboard a LEO satellite. Collecte Localisation Satellites (CLS) reports an ARGOS system accuracy of 150-2500 m depending on the scenario [2], however, Witt reports accuracies in the kilometer range on experiments tracking sea turtles as seen in Table 1.1 [32]. Table 1.1 organizes the localized tags by classes where each class corresponds to a tracking uncertainty. The highest number of estimates pertained to the classes with the highest uncertainty (A and B).

ARGOS Accuracy			
Location Class	N comparisons	CLS estimates [m]	Our Estimate [m]
3	54	< 250	400 ± 200
2	81	250-500	700 ± 700
1	133	500-1500	800 ± 700
0	108	> 1500	2300 ± 2700
A	398	No acc.	1400 ± 2500
B	546	No acc.	1800 ± 3900

Table 1.1: ARGOS tracking sea turtle accuracy

ARGOS was also tested to track movements of bottle nose dolphins and the final error was 7.4 kilometers on the total distance traveled, which is not reasonable [14]. Also, the cost of the ARGOS transmitting tag ranges from \$1500-\$4000 which is not feasible for large scale tracking [2], and the smallest ARGOS tag weighs 17g which is only appropriate for animals weighing 300 g or more. Applying GPS receivers on animals is another approach for tracking, however, these receivers are expensive, use large amounts of energy, and are also too heavy for smaller species. To address the cost, tracking accuracy, and weight of existing methods, we propose a new approach for wildlife tracking.

1.1 Contributions

In this thesis, we present a mission concept that relies on a constellation of low-cost cube satellites in LEO hosting receivers to detect and localize signals emitted from small transmitter tags on Earth. We compare the performance of a time of arrival measurement model and Doppler measurement model in terms of tag positioning root mean squared (RMS) error and a final state bias. Our simulation also includes necessary uncertainties such as the effect of the ionosphere, measurement noise, and satellite state estimate errors. We calculate the magnitude of the satellite state estimate uncertainty by implementing an extended Kalman filter (EKF) for a LEO satellite. Model mismatch in the filter is common in low earth orbiters due to the limited compute onboard, so we investigate bias modeling techniques for stochastic time processes to account for higher order gravity terms and other unknown forces. We culminate all the results into an interactive software tool to analyze and evaluate the effect the satellite constellation geometry and measurement accuracy have on the tag position accuracy.

1.2 Thesis Structure

The thesis is organized as follows: Chapter 2 describes the sequential estimation algorithm and implementation of orbit determination for LEO satellites. Chapter 3 covers the unified positioning algorithms using Doppler and time of arrival measurements with uncertainties as well as an end to end system architecture modeling and constellation geometry trade study. Chapter 4 provides a conclusion and future work on this mission concept. Finally, the Appendix covers important equation derivations in detail.

Chapter 2

Orbit Determination for Low-Earth Orbit Satellites using Sequential Estimation

2.1 Motivation

Obtaining an accurate state estimate of a spacecraft traversing along an orbit is essential for tracking because the tag position measurement functions such as time-of-arrival depend on the satellite state. Inaccurate satellite state estimates result in a greater tag state uncertainty. Using GPS measurements as the satellite state estimate is infeasible because of sensor inaccuracies due to hardware limitations, and only using a dynamics model is also impractical as these set of differential equations serve as an approximation of the real environment. For this project, we utilize a truncated dynamics model to save computation onboard, therefore unmodeled effects prevent a forward simulation of the dynamics to be used as the state estimate. In this chapter, we explore sequential state estimation for satellites in LEO to provide a maximum likelihood state estimate at each timestep along a reference orbit. We discuss the extended Kalman filter (EKF) algorithm in detail as well as implementation details such as covariance tuning and bias modeling for unmodeled dynamics.

2.2 Background

2.2.1 Gaussian Distributions and Covariance

Gaussian (normal) probability density functions are used to describe continuous random variables in sequential state estimators like the Kalman Filter. These functions are only characterized by the mean of the random variable (μ) and a variance (σ^2). The mean is the expected value ($E(X)$) of the distribution while the variance is a metric that describes the spread of possible values the random variable can take. This spread is measured with respect to the mean and it is essentially the mean squared error about the mean.

It is important to note that the square root of the variance is the standard deviation (σ) which is another metric that measures the variability of the random variable from the mean. It is expected that 68.2% of all the values the random variable can take are within 1 standard deviation away from the mean on both sides. The percentages for two-sigma and three-sigma are 95% and 99.7% respectively.

Equation (2.1) is the equation for a univariate Gaussian and Figure 2.1 shows this distribution.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{1}{2}(\frac{x-\mu}{\sigma})^2) \tag{2.1}$$

For state estimation, most states are multidimensional, therefore a parameter to describe the relationship between all the state variables is needed. This parameter is known as the covariance matrix (Σ) which is symmetric positive definite and the diagonals consist of the variances of each element in the state while the off diagonals capture the relationship between all possible pair of variables in the state. A n-dimensional multivariate Gaussian can be viewed as a collection of n independent distributed Gaussian distributed random variables with means μ_i and variances σ_i^2 [9]. The equation for a multivariate Gaussian is shown in Equation (2.2)

$$f(x) = \frac{1}{|\Sigma|^{1/2}(2\pi)^{n/2}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)) \tag{2.2}$$

One of the favorable properties of Gaussians is once they are input into a linear

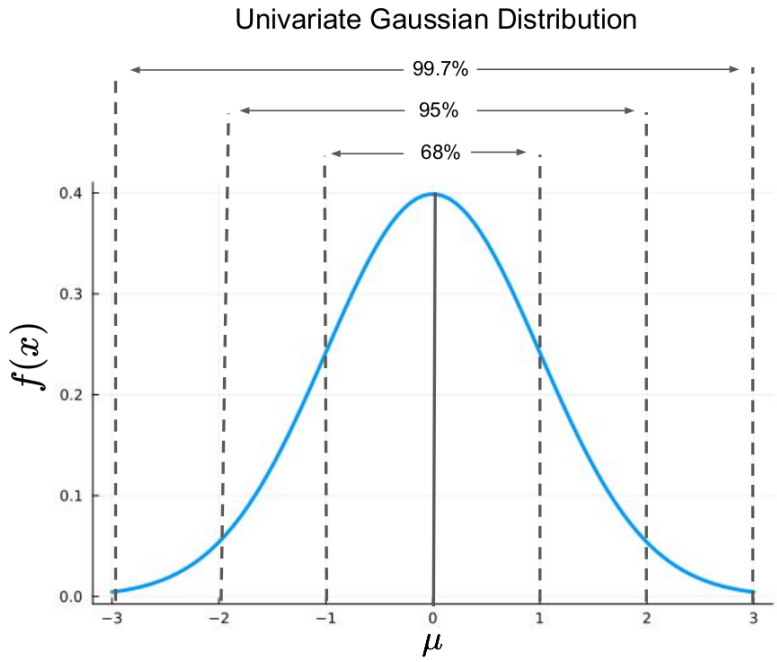


Figure 2.1: Univariate Gaussian distribution with zero mean and a standard deviation equal to 1. The area under the curve denotes the probability of the continuous random variable to be a value between the integration interval.

function, the output is also a Gaussian. Also the central limit theorem states that for an infinite number of samples of size n from an arbitrary probability density function, the mean of each of those n samples is normally distributed. The notation used throughout this thesis is shown in Equation (2.3).

$$x \sim \mathcal{N}(\mu, \Sigma) \tag{2.3}$$

Equation (2.3) means that the multidimensional variable x is sampled from a Gaussian distribution with mean μ and covariance Σ.

2.2.2 Satellite Dynamics

The satellite is modeled as a point mass that experiences a wide range of accelerations in LEO. The sources of these accelerations include Earth's gravity field, Earth's atmosphere, solar radiation pressure (srp), and additional forces caused by the

gravitational pull from third bodies such as the sun and moon. The details of these accelerations are described in the following subsections. The state of the satellite is represented by a six-dimensional vector ($\vec{x} \in R^6$) in Cartesian coordinates which is composed of the satellite position with respect to the center of the Earth ($\vec{r} \in R^3$) and the satellite velocity ($\vec{v} \in R^3$). Both of these quantities are described in the Earth Centered Inertial (ECI) frame. The sum of all these accelerations is the ground truth satellite dynamics model that is used in the state estimator. This ground truth model is used to test the consistency of the filter along with the accuracy of the estimator.

Earth Gravity

The effect of the gravitational force can be derived from a potential since it is a conservative force. The potential for Earth's gravity (U) is an infinite summation of perturbations which is described by Equation (2.4).

$$U = \frac{GM_{\oplus}}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \frac{R_{\oplus}^{n}}{r^{n}} P_{nm}(sin\phi)(C_{nm}cos(m\lambda) + S_{nm}sin(m\lambda)) \tag{2.4}$$

In this equation, G represents the gravitational constant, $M_{\oplus}$ is the mass of Earth, r is the scalar distance of the satellite from the center of the Earth in ECI coordinates, ϕ and λ represent latitude and longitude of the spacecraft, C_{nm} and S_{nm} are geopotential coefficients that are related to the mass distribution of Earth, and P_{nm} is the associated Legendre polynomial of degree n and order m. The geopotential coefficients are obtained through a table that was generated through gravity model improvement missions [24].

The most dominant higher-order gravity force is known as J_2. This force emerges due to the fact that the Earth is not a perfect sphere but rather an oblate spheroid where the equatorial radius exceeds the polar diameter by approximately 20 km [24]. This difference in diameter exerts a force on the spacecraft that attempts to align the orbital plane with the equator. A derivation of the gravitational potential for up to J_2 is shown in the Appendix. In summary, the infinite series in (2.4) describes the effect of the perturbations on the gravity field and it is dependent on latitude, longitude, Legendre polynomials, and geopotential coefficients.

Once the gravity potential is derived to the desired order and degree, the acceler-

ation is determined by Equation (2.5).

$$a_g = \nabla U \tag{2.5}$$

A closed form equation in of Earth's gravitational force up to J_2 in Cartesian coordinates is shown in Equation (2.6).

$$a_{gJ2} = \frac{-\mu}{r^2} + \frac{3\mu J_2 R_\oplus^2}{2||\vec{r}||^5}\left(\frac{5(\vec{r}\cdot\vec{I_z})^2}{||\vec{r}||^2} - 1\right)\vec{r} - 2(\vec{r}\cdot\vec{I_z})\vec{I_z} \tag{2.6}$$

In Equation (2.6), μ is the standard gravitational parameter which is equivalent to $GM_\oplus$, J_2 is known as the zonal coefficient which is equivalent to $-C_{nm}$ with n=2 and m=0. Setting m=0 means that this force does not depend on longitude of the spacecraft. The variable $\vec{I_z}$ corresponds to the unit vector pointing in the z direction.

2.2.3 Atmospheric Drag

The spacecraft also experiences a drag force in LEO due to the resistance of Earth's atmosphere. This force acts in the opposite direction of the velocity vector and is characterized by Equation (2.7).

$$a_d = -\frac{1}{2}C_D A\rho||\vec{v_{rel}}||\vec{v_{rel}} \tag{2.7}$$

In Equation (2.7), C_D represents the drag coefficient which depends on the interaction of the satellite surface material and the atmosphere, ρ is the atmospheric density which is obtained through the Harris-Priester model [24], A is the cross sectional area perpendicular to the velocity vector, and v_{rel} is the relative velocity of the satellite with respect to the atmosphere and Equation (2.8) shows this relation.

$$v_{rel} = \vec{v} - \omega_{EARTH} \times \vec{r} \tag{2.8}$$

For this simulation, the drag coefficient and cross sectional area were set arbitrarily and assumed to be constant throughout orbit.

Solar Radiation Pressure

The eccentricity of the satellite orbit is affected by solar radiation pressure from the sun and it is proportional to the surface area and mass of the satellite. For this project, we assume the mass of the satellite to be 1 kg. This force is due to the reflection and absorption of the photons, and it is characterized by Equation (2.9) with the assumption that the surface normal points toward the sun [24].

$$a_{srp} = -P_{\odot} C_R \frac{A}{m} \frac{\vec{r_{\odot}}}{||\vec{r_{\odot}}||^3} (AU)^2 \tag{2.9}$$

In Equation (2.9), $P_{\odot}$ is the solar radiation pressure, A is the cross sectional area in the direction of the sun, $r_{\odot}$ is the vector from the spacecraft to the sun, and AU is an astronomical unit.

Third Body Forces

The sun and moon also exert forces on the spacecraft and they are modeled using Newtons law of gravity shown in Equations (2.10) and (2.11).

$$a_{sun} = \frac{GM_s(\vec{r}_{sun} - \vec{r})}{||\vec{r}_{sun} - \vec{r}||^3} \tag{2.10}$$

$$a_{moon} = \frac{GM_m(\vec{r}_{moon} - \vec{r})}{||\vec{r}_{moon} - \vec{r}||^3} \tag{2.11}$$

In these equations, M_s is the mass of the sun and M_m is the mass of the moon. Similarly, r_{sun} represents the position of the sun in ECI frame and r_{moon} is the position of the moon in the ECI coordinate frame.

2.3 Extended Kalman Filter

The discrete extended Kalman filter (EKF) is a sequential state estimation algorithm that optimally fuses imperfect measurements and an uncertain nonlinear dynamics model to obtain the best state estimate of a system. In this algorithm, the observations are processed individually as they are measured by the spacecraft. The EKF is

a variant of the Kalman filter to account for nonlinear dynamics and nonlinear measurement functions which is the case for a spacecraft. The nonlinear dynamics are linearized about the current estimated mean and the measurement function is linearized about the propagated mean. The mean is modeled as a belief which is a random variable characterized by a Gaussian distribution, and the uncertainty is represented through a covariance matrix. The purpose of a Kalman filter is to propagate a Gaussian forward in time through a linear transformation and calculate the new mean and covariance in closed form. These values correspond to the next exact belief. However, with a nonlinear dynamics model, only an approximation of the belief can be made. This is because of the linear approximation via the first-order Taylor expansion of the nonlinear dynamics. The quality of the linear approximation depends on the uncertainty of the dynamics as well as the nonlinearities that exist in the function at the linearization point.

To validate the filter is converging, a ground truth model must be defined. For this application, the ground truth model is a summation of all the forces described in Section 2.2.2 and it is shown in Equation (2.12). The gravitational force (a_g) consists of higher-order gravity terms up to degree and order 10.

$$a_{gt} = a_g + a_d + a_{sun} + a_{moon} + a_{srp} \tag{2.12}$$

The EKF consists of two primary steps: a prediction step where and a correction step and both will be discussed in detail.

2.3.1 Prediction Step

The nonlinear dynamics model used in the filter is known as the process model and it only consists of two body gravity force with J2 as well as the drag acceleration. The entire dynamics are not included in the filter to reduce computation in each step as the EKF requires the computation of a discrete dynamics jacobian at each timestep. Equation (2.13) shows the process model dynamics.

$$a_{process} = a_{gJ2} + a_d \tag{2.13}$$

Equations (2.12) and (2.13) are continuous dynamics functions, and we will define

notation for the discrete versions of these models. The discrete ground truth model is defined as $x_{k+1} = f_{gt}(x_k)$ and the discrete process model is defined as $x_{k+1} = f_p(x_k)$. The discrete models are obtained by numerically integrating the continuous dynamics models using a fourth-order Runga Kutta (RK4) integrator. Equation (2.14) shows the relationship we assume between the state at time $k+1$ of the discrete ground truth model and state at time k of the discrete process model.

$$x_{k+1} = f_p(x_k) + q_k \tag{2.14}$$

The term q_k corresponds to additive white Gaussian noise (AWGN) which means q_k is sampled from a zero mean Gaussian distribution with covariance matrix Q (process noise covariance matrix) as shown in Equation (2.15). Additive white noise requires a constant standard deviation with time along with no correlation between time steps.

$$q_k \sim \mathcal{N}(0,\, Q) \tag{2.15}$$

The unmodeled accelerations and higher-order acceleration terms from the gravity geopotential are treated as process noise for some versions of the filter. A set of residuals shown in Equation (2.16) are calculated until a final simulation time (T) and their covariance is used to generate an initial guess for the process noise covariance. The variable $\bar{x}_k$ is the reference ground truth trajectory from a forward roll out on the ground truth dynamics. The final algorithm required additional hand tuning to the process noise covariance matrix for a consistent result.

$$r_k = f_p(\bar{x}_k) - \bar{x}_{k+1} \tag{2.16}$$

In the prediction step, the predicted mean (x_{k+1}^-) is calculated by propagating the corrected mean (x_k^+) forward in time using the nonlinear discrete deterministic process model dynamics, and the predicted covariance is calculated using the discrete process model dynamics jacobian (A_k) and the process noise covariance (Q). The prediction step is summarized in Equation (2.17).

$$\begin{aligned} x_{k+1}^- &= f_p(x_k^+) \\ P_{k+1}^- &= A_k P_k^+ A_k^T + Q \end{aligned} \tag{2.17}$$

2.3.2 Correction Step

The Kalman filter also requires a measurement prediction dependent on the predicted mean. For this mission, we assume that the spacecraft has a NovAtel GPS receiver onboard. The measurement model is simplified by assuming it is linear function that outputs position of the spacecraft, however additive noise is included in the simulation as shown in Equation (2.18).

$$\begin{aligned} z_k &= h(x_k) + r_k \\ h(x_k) &= Cx_k \\ C &= [I_{3x3} \quad 0_{3x3}]^T \end{aligned} \tag{2.18}$$

The measurement noise (r_k) is also assumed to be AWGN and it is sampled from a zero mean Gaussian distribution with covariance R (Equation (2.19)). The magnitude of the elements in R is determined by the accuracy of the measurement which is usually provided on a sensor specification sheet. We approximate the accuracy of GPS to be about 10 meters [25], so we construct R matrix according to this metric. All the implementation details will be described in Section 2.3.6.

$$r_k \sim \mathcal{N}(0,\, R) \tag{2.19}$$

The correction step incorporates the information from a measurement at timestep $k+1$ to correct the prediction. This step also includes the calculation of the Kalman gain K_{k+1}, the corrected mean x^+_{k+1}, and the corrected covariance P^+_{k+1}. A derivation of the Kalman filter equations is shown in [29], and this section will describe the overall intuition of each calculated parameter. Equation (2.20) shows the calculation of the Kalman gain. This parameter captures the ratio of the uncertainty in prediction over the uncertainty of the predicted measurement error. The predicted measurement error is the difference between the true measurement at timestep $k+1$ (z_{k+1}) and the predicted measurement. It is used as a scale factor to determine how much of the predicted measurement error to use to update the predicted mean. A high Kalman gain means that the measurements are accurate and the prediction is inaccurate, and a low Kalman gain allows the filter to trust the predictions more than the measurements.

$$K_{k+1} = P_{k+1}^{-} C^T (C P_{k+1}^{-} C^T + R)^{-1} \tag{2.20}$$

The mean correction step is shown in Equation (2.21) and the covariance correction is described in (2.22).

$$x_{k+1}^{+} = x_{k+1}^{-} + K_{k+1}(z_{k+1} - h(x_{k+1}^{-})) \tag{2.21}$$

$$P_{k+1}^{+} = (I - K_{k+1}C)P_{k+1}^{-}(I - K_{k+1}C)^T + K_{k+1}RK_{k+1} \tag{2.22}$$

2.3.3 Assumptions

One of the assumptions the EKF makes is that the dynamics and measurement functions are differentiable. This is because the algorithm requires derivatives of these functions for the linear approximation. Another assumption is that the uncertainty of the state is centered around the mean. Lastly, the state difference between the process model and the ground truth model have to be additive white Gaussian noise. For this application, the higher-order gravity terms and additional unmodeled accelerations are technically not white as the standard deviation changes between time steps which results in a non optimal state estimate [33]. This causes the state difference between the process and ground truth model to also violate the assumption. Figure 2.2 shows the magnitude of the unmodeled accelerations in each dimension over a 5 period horizon. Alternate filter implementations estimate the additional accelerations by adding it to the state. More detail on adding the unmodeled accelerations to the state are in Section 2.3.4.

However, if the ground truth model and the process model have the same order gravity terms, the unmodeled accelerations behave in a sinusoidal pattern and are smooth which make them easier to estimate out in the filter. Figure 2.3 shows the unmodeled accelerations once the ground truth and process model have the same order gravity terms.

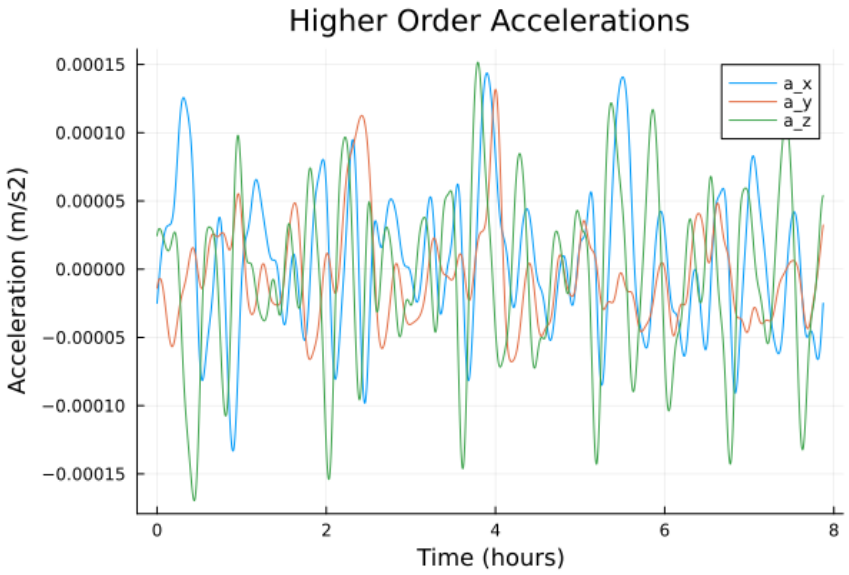


Figure 2.2: Unmodeled accelerations in LEO between the ground truth model with order 10 gravity and the process model with up to J2

2.3.4 Bias Modeling

The errors between the process model and ground truth model will eventually cause the filter to diverge or be inconsistent if the process noise is not tuned correctly. In the previous section, the addition of process noise is known as state noise compensation (SNC) and it attempts to correct the model difference [5]. However, this assumes that the state difference due to the unmodeled accelerations behave as a white Gaussian process which is not the case. Figure 2.2 shows that the unmodeled accelerations are time correlated because the standard deviations at each timestep are not the same throughout the entire simulation. We obtain the effect the unmodeled accelerations have on the state through integration, and these effects are also not white. We then attempt to estimate out an approximation of the unmodeled accelerations ($\varepsilon \in R^3$) by adding them to the state. This approach is known as dynamic model compensation (DMC) and it has been shown to provide more precise state estimates for lunar satellites using Apollo 10 data [28] and near-Earth orbiters [22]. Model errors are always present because of uncertainties in atmospheric density models, variations in solar radiation pressure, and tidal forces, etc [22]. Our approach only corrects the model errors arising from the truncated process dynamics model.

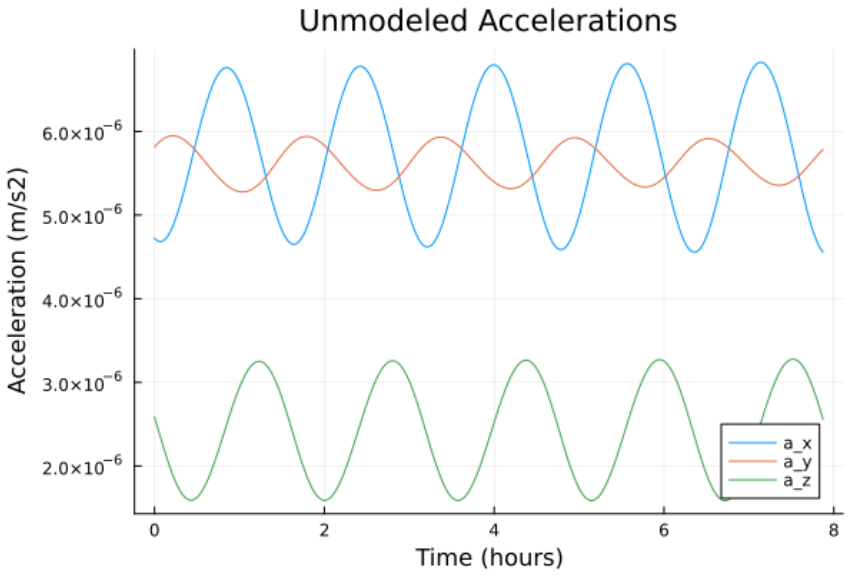


Figure 2.3: Unmodeled accelerations in LEO for a ground truth model and a process model with the same order gravity terms

One approach is to assume the accelerations follow a stochastic process known as a random walk which is modeled in Equation (2.23).

$$\begin{aligned} \dot{\varepsilon} &= w(t) \\ w(t) &\sim \mathcal{N}(0,\, Q_\varepsilon) \end{aligned} \tag{2.23}$$

The covariance matrix Q_ε is manually set depending on how much variation one predicts the unmodeled accelerations to have.

Another approach is to use a first-order, Gauss-Markov stochastic process in Equation (2.24) to describe the evolution of the unmodeled accelerations over time [22]. Estimating out these unmodeled accelerations can be used to improve the current truncated dynamics model, however the disadvantage is an increase in computation due to the size of the EKF state increasing [6].

$$\begin{aligned} \dot{\varepsilon} &= -B(t)\varepsilon + w(t) \\ w(t) &\sim \mathcal{N}(0,\, Q_\varepsilon) \end{aligned} \tag{2.24}$$

The 3 parameter vector $w(t)$ is random noise sampled from a Gaussian distribution

with covariance Q_ε and B is a diagonal 3x3 matrix of the time correlation coefficients $\beta_x, \beta_y, \beta_z$. The time correlation coefficients are also set to follow a random walk in Equation (2.25).

$$\begin{aligned} \dot{\beta} &= u(t) \\ u(t) &\sim \mathcal{N}(0,\, Q_\beta) \end{aligned} \tag{2.25}$$

The new state becomes a twelve-dimensional vector that includes the position, velocity, unmodeled accelerations, time correlation coefficients. The implementation details are provided in Section 2.3.6.

The drag force is difficult to model due to the complexity of the atmosphere as it experiences variations due to solar activity and Earth's magnetic field. For example, [31] reports the maximum atmospheric density is $5e^{-11}\frac{kg}{m^3}$ and the minimum is $2e^{-14}\frac{kg}{m^3}$ at an altitude of 500 km given solar flares. This difference of 3 orders of magnitude is not predictable in a atmospheric density model like the Harris-Priester model, therefore estimating out the drag term in the filter is another option for filter design. One approach is to estimate the drag term plus the unmodeled accelerations in one term and the other is to estimate a scale factor (k_ρ) from the atmospheric density obtained from the Harris-Priester (ρ_{est}) model as shown in (2.26).

$$\rho_{true} = k_\rho \rho_{est} \tag{2.26}$$

The atmospheric density is not estimated directly because the magnitude of the this term is approximately order $1e^{-12}$ which is drastically smaller than the position and velocity of the spacecraft. This difference in magnitude leads to ill conditioning in the filter covariance updates. However, estimating out the scale factor improves the numerical properties of the filter because this value is approximately order 1.

2.3.5 Square Root Filtering

Throughout the correction of the covariance matrix, round off errors may occur due to poorly conditioned problems. This may cause the matrix to not be positive semi definite which is a problem for the Kalman filter. One reason this can occur is when the covariance matrix is rapidly reduced by processing accurate measurements [16].

Square root filtering avoids this problem by propagating the matrix square root (F) of the covariance matrix (P) as shown in Equation (2.27). For this implementation, the square root of a semi positive definite matrix is defined as the lower Cholesky factor of the covariance matrix. This improves the numerical properties of the problem as it doubles the amount of working precision which allows the algorithm to maintain the positive semi definite nature of the covariance matrix [16].

$$P = FF^T \tag{2.27}$$

The square root filter requires the ability of expressing the square root of a sum of matrices as a function of the individual square roots [30]. This is obtained by leveraging the properties of the QR factorization, as the square root of the sum of matrices is the R matrix of the QR factorization of each individual square root matrix concatenated and this relation is shown in Equation (2.28).

$$\begin{aligned} Q, R &= qr(\begin{bmatrix} \sqrt{A} \\ \sqrt{B} \end{bmatrix}) \\ R &= \sqrt{A+B} \end{aligned} \tag{2.28}$$

The property in Equation (2.28) is used to reformulate the original extended Kalman filter update equations. The final square root extended Kalman filter algorithm is shown in Algorithm 1. In this algorithm, μ is the mean of the estimate, F is the square root of the covariance matrix, A is the discrete process dynamics jacobian evaluated at the corrected mean, C is the linear measurement function matrix, f_p is the nonlinear discrete deterministic process dynamics model, Γ_W is the square root of the process noise covariance, and Γ_V is the square root of the measurement noise covariance.

2.3.6 Implementation and Results

Various versions of the EKF were implemented to conduct a filter design analysis for low-earth-orbiting satellites. Each version differs in the assumptions made between the ground truth model and the process model as well as the state that is being estimated. The process model described in Section 2.3.1 is used in all the versions

Algorithm 1 Square Root Extended Kalman Filter using QR factorization

function QR EKF($\mu_{t|t}, F_{t|t}, y_{t+1}, A, C, \Gamma_W, \Gamma_V$)
- $\mu_{t+1|t} = f_p(\mu_{t|t})$ ▷ State Prediction
- $\_, F_{t+1|t} = qr(F_{t|t}A^T, \Gamma_W)$ ▷ Covariance Prediction
- $z = y_{t+1} - C\mu_{t+1|t}$ ▷ Measurement Innovation
- $\_, G = qr(F_{t+1|t}C^T, \Gamma_V)$ ▷ Innovation Covariance
- $K = [G^{-1}(G^{-T}C)F_{t+1|t}^T F_{t+1|t}]^T$ ▷ Kalman Gain
- $\mu_{t+1|t+1} = \mu_{t+1|t} + Kz$ ▷ State Update
- $F_{t+1|t+1} = qr(F_t + 1|t(I - LC)^T, \Gamma_V L^T)$ ▷ Covariance Update

and a ground truth trajectory was generated by simulating the nonlinear ground truth dynamics forward for 5 orbits. The initial condition for the forward roll out corresponds to the polar orbit chosen for the wildlife-tracking mission. The orbital parameters for the wildlife-tracking orbit are shown Table 2.1.

Orbital Parameters	
Semimajor Axis	6871e3
Eccentricity	0.0004879
Inclination	90.6391
RAAN	194.5859
Argument of Perigee	151.2014
Mean Anomaly	190

Table 2.1: Orbital parameters for the ground truth model

All the simulations were run in Julia [4], and the open-source Julia package *SatelliteDynamics.jl* [11] was used to simulate ground truth orbit. Filter performance was based on ensuring the difference between the estimated EKF trajectory and the ground truth trajectory (residual) is within the three-sigma covariance bound of the EKF covariance matrix over time. If the errors are outside of these bounds, the filter believes the estimate is accurate, however the estimated state is diverging from the ground truth. This is known as filter inconsistency and it is caused due to the filter being too optimistic. A filter is optimistic when the measurements or process model are assumed to be better than they actually are. Tuning the process noise covariance was essential for this implementation to converge, and this was done through checking the consistency of the filter after several simulation runs.

The metric to determine the accuracy of the estimated state is the root mean squared (RMS) error between the estimated trajectory (x_{est}) and the ground truth trajectory ($\bar{x}$). Equation (2.29) shows the calculation of the RMS error, and we compute this error on the position, velocity, and unmodeled accelerations for each version. The variable k represents the each timestep along the trajectory, and K is the total number of time steps in the horizon.

$$RMSE = \sqrt{(\frac{\sum_{k=1}^{K} ||x_{est_k} - \bar{x}_k||^2}{K})} \tag{2.29}$$

The results are summarized in Table 2.2. The final estimated covariance matrix can also be used to calculate the RMS error, however this value measures how accurate the filter believes it is. We label this metric as $RMSE_\Sigma$. Since we use the square root variant of the EKF, we must obtain the original covariance by multiplying the covariance square root (F) by it's transpose. We then take the square root of the trace (tr) corresponding to the position elements $(1:3, 1:3)$ to obtain ($RMSE_\Sigma$). The same metric is used for the velocity and unnmodeled acceleration estimates, and the only change is the indices of the covariance matrix in Equation (2.30).

$$RMSE_\Sigma = \sqrt{tr((FF^T)_{1:3,1:3})} \tag{2.30}$$

The RMS error from the final covariance matrix of the EKF should be within close range of the RMS error calculated from the residuals in Equation (2.29). This ensures the filter is consistent and not diverging from the ground truth.

The overall accuracy of the state estimate is best when the filter is optimistic, however the estimated trajectory diverges from the ground truth trajectory over time. Increasing the variances in the process noise covariance matrix prevents the filter from being being optimistic. The trade off in Kalman filters is consistency versus accuracy, and tuning the process noise affects this relation. The measurement noise for all implementations was set to reflect a GPS position uncertainty to be 10 meters. The derivation and measurement noise covariance matrix is shown in (2.31).

$$\begin{aligned} Var(x+y+z) &= 10^2 \\ Var(x)+Var(y)+Var(z) &= 100 \\ R &= \left[\tfrac{100}{3} I(3)\right] \end{aligned} \tag{2.31}$$

The following subsections will describe each version of the EKF along with errors of each implementation.

EKF Errors			
Version	Position RMS Error (m)	Velocity RMS Error (m/s)	Acc. RMS Error (m/s^2)
1	1.352	0.0158	N/A
2	0.692	0.00146	1.676e-6
3	1.345	0.0125	8.152e-5
4	1.237	0.0127	9.548e-5

Table 2.2: EKF Version Errors

Estimator with SNC and different order gravity models

In this implementation, we use a ground truth model up to order and degree 10. The state of the filter is six-dimensional and includes the position and velocity of the spacecraft in ECI coordinates ($x = [r\ v]$). Figures 2.4, 2.5, and 2.6 show the position residuals and Figures 2.7, 2.8, 2.9 illustrate the velocity residuals along with the three-sigma bounds over a simulation time of 5 orbits. The filter is consistent as these residuals are bounded by the covariance bounds, and the position RMS error is 1.352 meters. This version of the filter treats the higher-order gravity terms and the unmodeled accelerations from solar radiation pressure and acceleration from the sun and moon as AWGN, and the process noise covariance matrix for this version is shown in (2.32).

$$Q_{v_1} = \begin{bmatrix} (1\times10^{-2})I(3) & 0 \\ 0 & (8.2\times10^{-4})I(3) \end{bmatrix}^2 \tag{2.32}$$

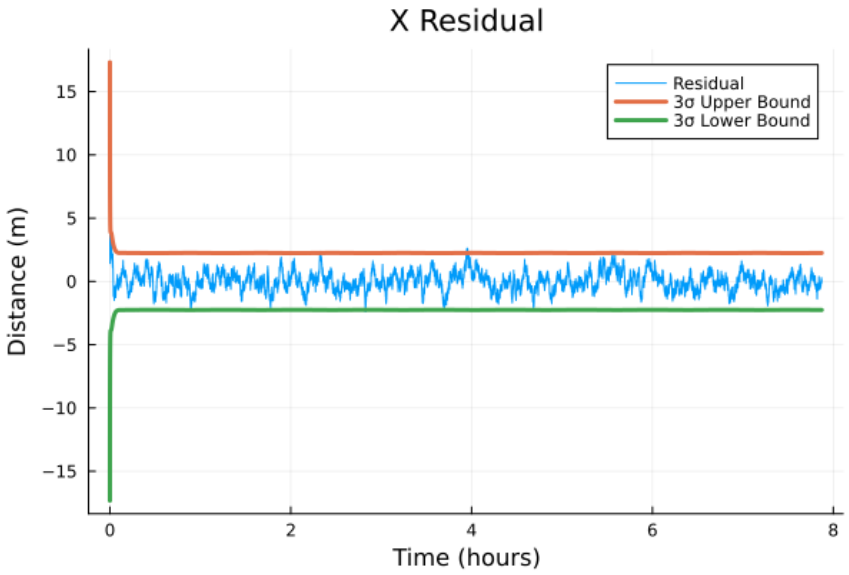


Figure 2.4: Estimator with SNC and different order gravity models X position residual

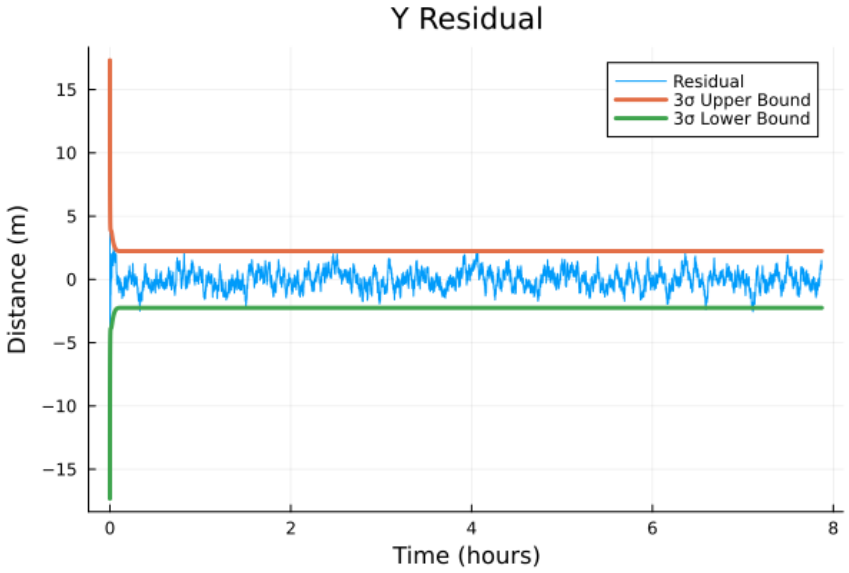


Figure 2.5: Estimator with SNC and different order gravity models Y position residual

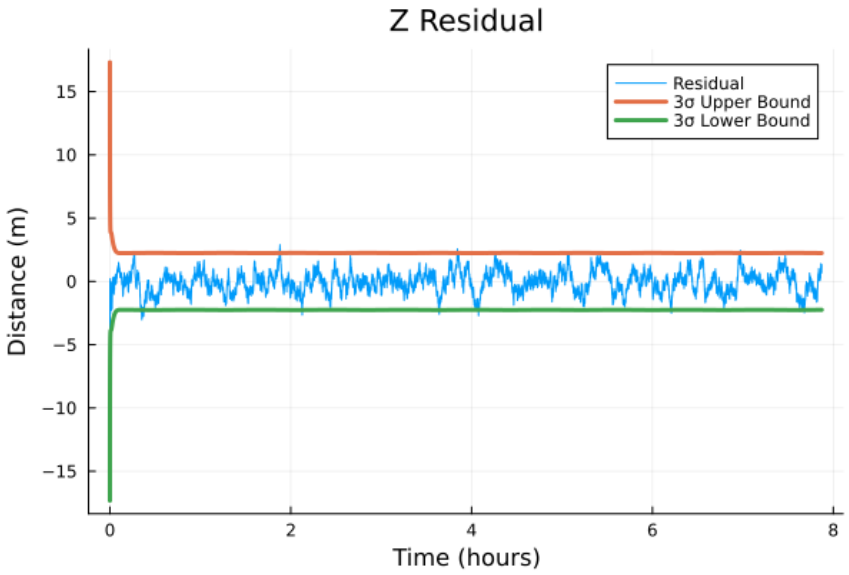


Figure 2.6: Estimator with SNC and different order gravity models Z position residual

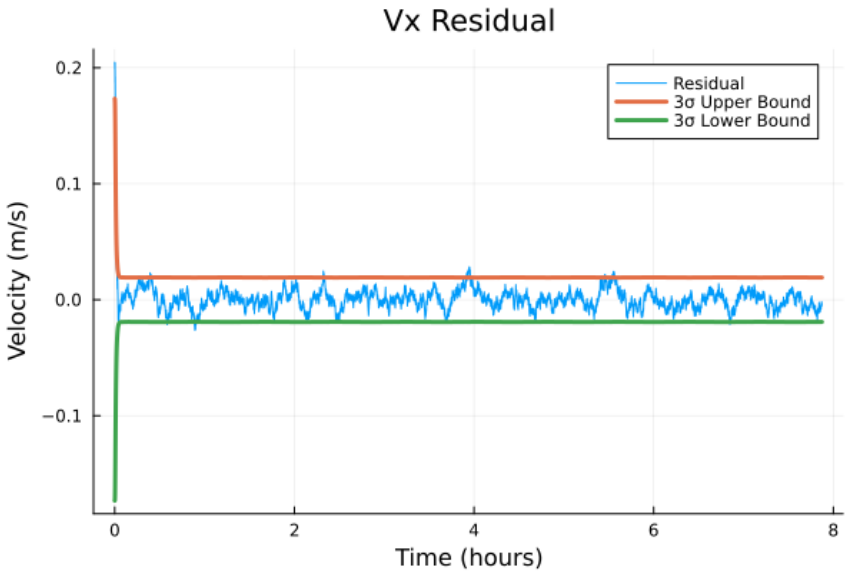


Figure 2.7: Estimator with SNC and different order gravity models X velocity residual

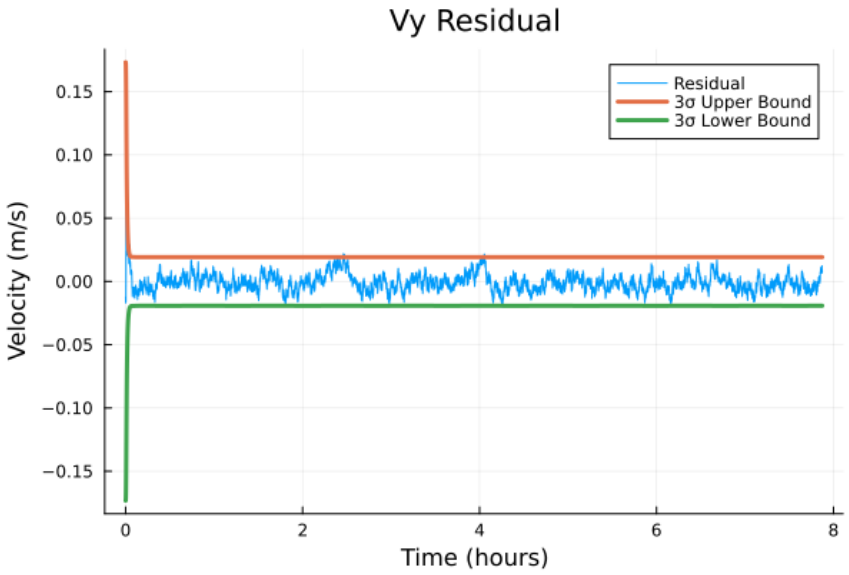


Figure 2.8: Estimator with SNC and different order gravity models Y velocity residual

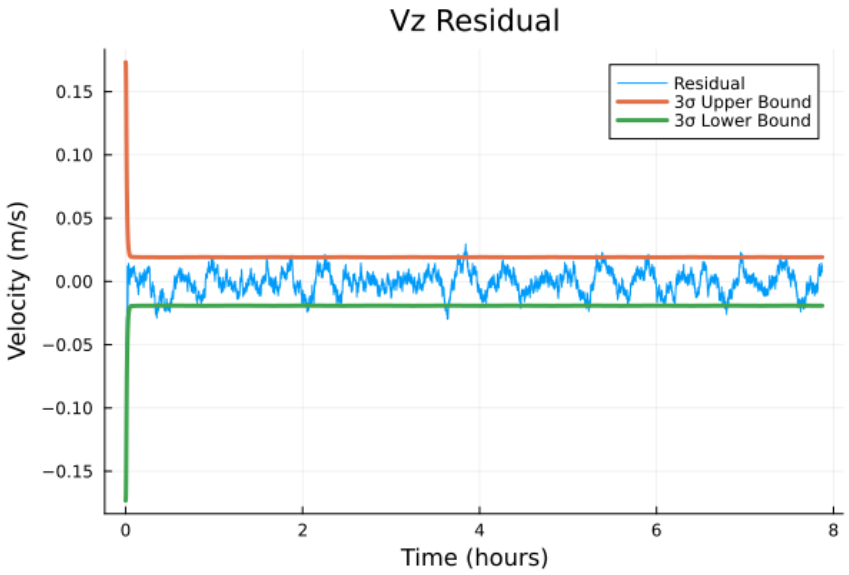


Figure 2.9: Estimator with SNC and different order gravity models Z velocity residual

Estimator with DMC via random walk and the same order gravity models

This version assumes that the process model and the ground truth model have the same order gravity terms. Both were modeled up to J2 and the unmodeled accelerations consist of forces from solar radiation pressure, the sun, and the moon. However, these accelerations behave smoothly as seen in Figure 2.3, and this is not the case with the higher-order gravity terms. The smooth sinusoidal behavior makes it more accessible for the filter to estimate out these additional acceleration terms, however it still is not AWGN, so a position error is still expected. The state for this version is the spacecraft position, velocity, the unmodeled accelerations modeled by a random walk, and the atmospheric density scale described in Section 2.3.4. The position residuals are shown in Figures 2.10, 2.11, and 2.12, and the velocity residuals are displayed in Figures 2.13, 2.14, and 2.15. Both sets of residual plots are consistent for the majority of the simulation time. Additionally the unmodeled acceleration residuals are shown in Figures 2.16, 2.17, and 2.18, and these plots highlight the filter is more pessimistic in the acceleration terms compared the others states. Overall, this version had the best performance with a position accuracy of 0.692 m, however the assumptions made are not realistic as the higher-order terms do effect the spacecraft trajectory significantly over long horizons. The process noise is shown in 2.33.

$$Q_{v_2} = \begin{bmatrix} (1.1547 \times 10^{-5})I(3) & 0 & 0 & 0 \\ 0 & (5.1962 \times 10^{-5})I(3) & 0 & 0 \\ 0 & 0 & (3 \times 10^{-8})I(3) & 0 \\ 0 & 0 & 0 & 5 \times 10^{-4} \end{bmatrix}^2 \tag{2.33}$$

The algorithm was also able to successfully track the atmospheric scale consistently as shown in Figure 2.19.

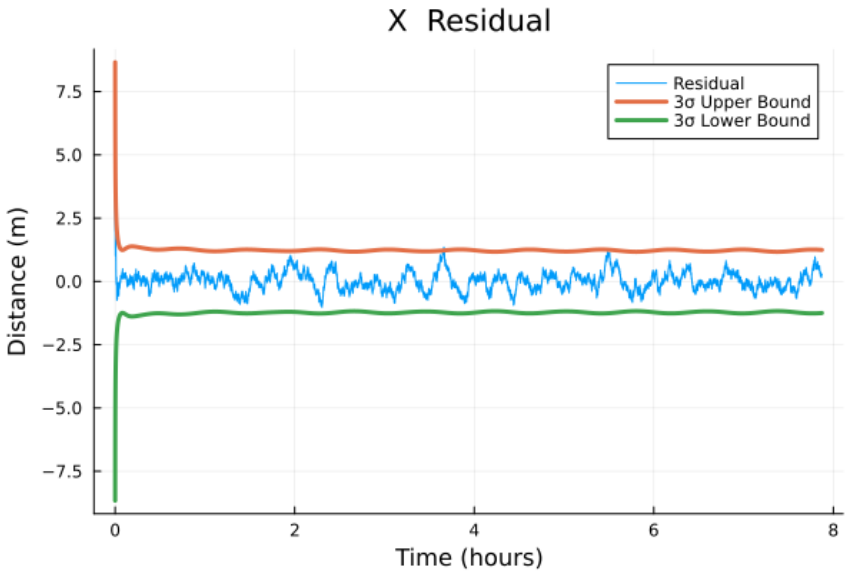


Figure 2.10: Estimator with DMC via random walk and the same order gravity models X position residual

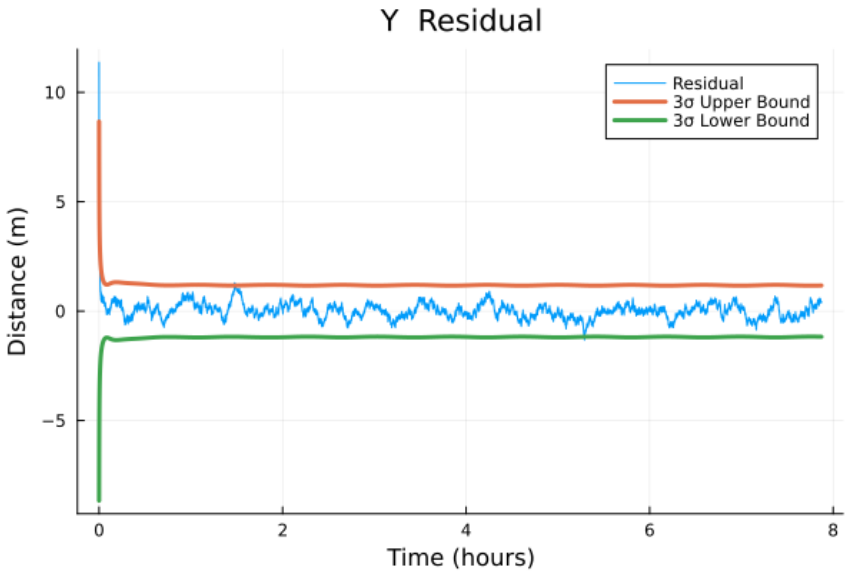


Figure 2.11: Estimator with DMC via random walk and the same order gravity models Y position residual

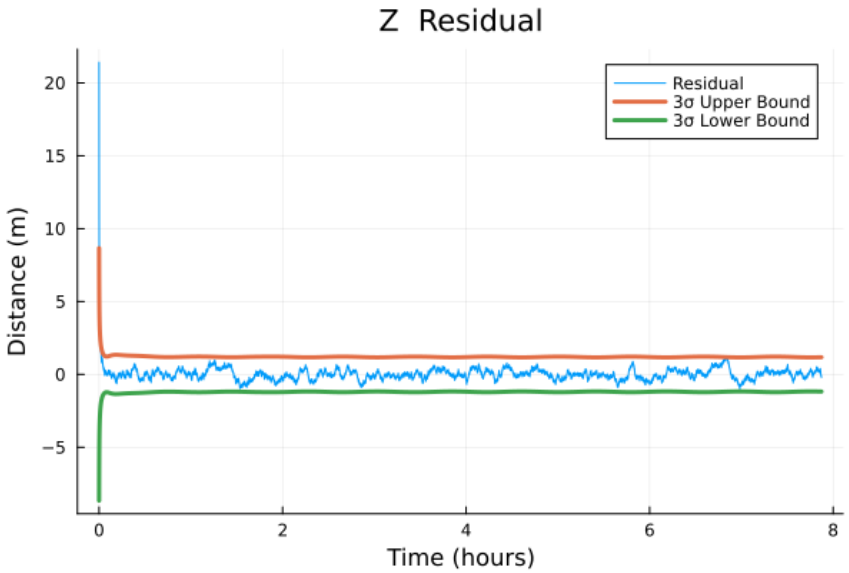


Figure 2.12: Estimator with DMC via random walk and the same order gravity models Z position residual

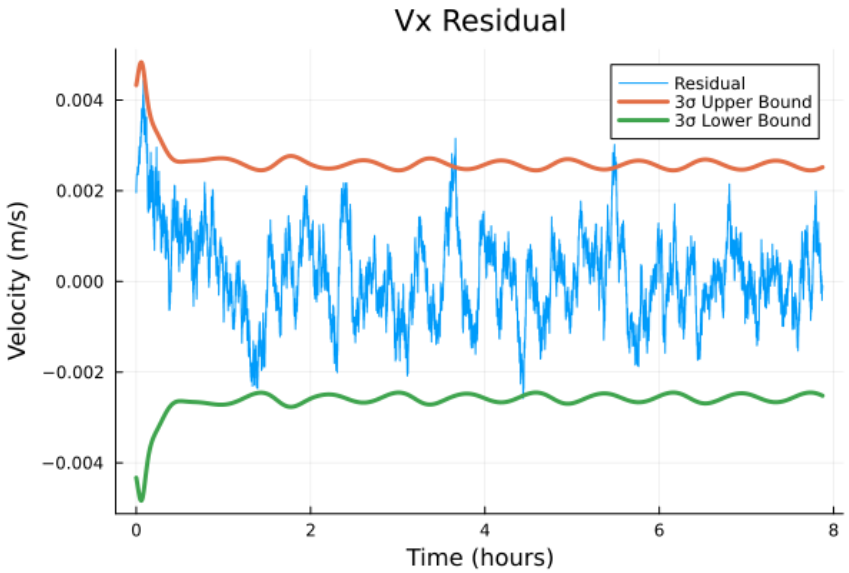


Figure 2.13: Estimator with DMC via random walk and the same order gravity models X velocity residual

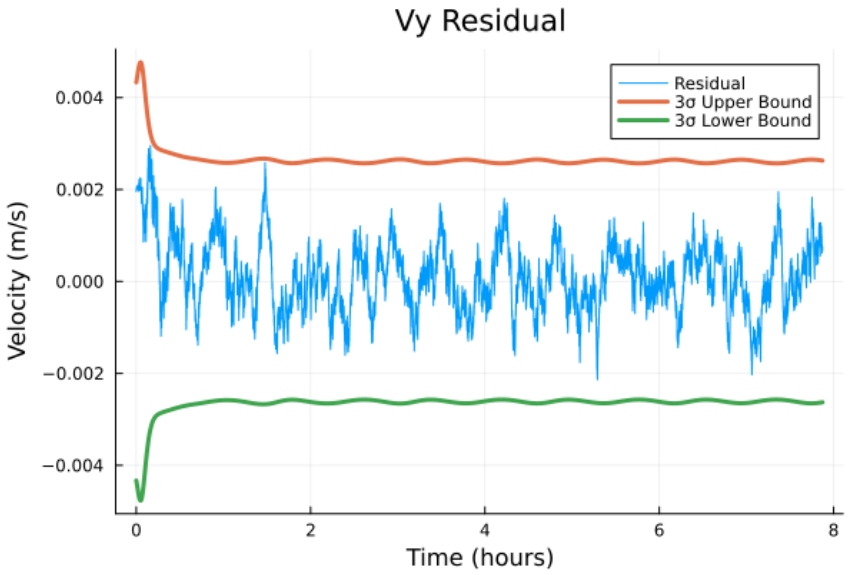


Figure 2.14: Estimator with DMC via random walk and the same order gravity models Y velocity residual

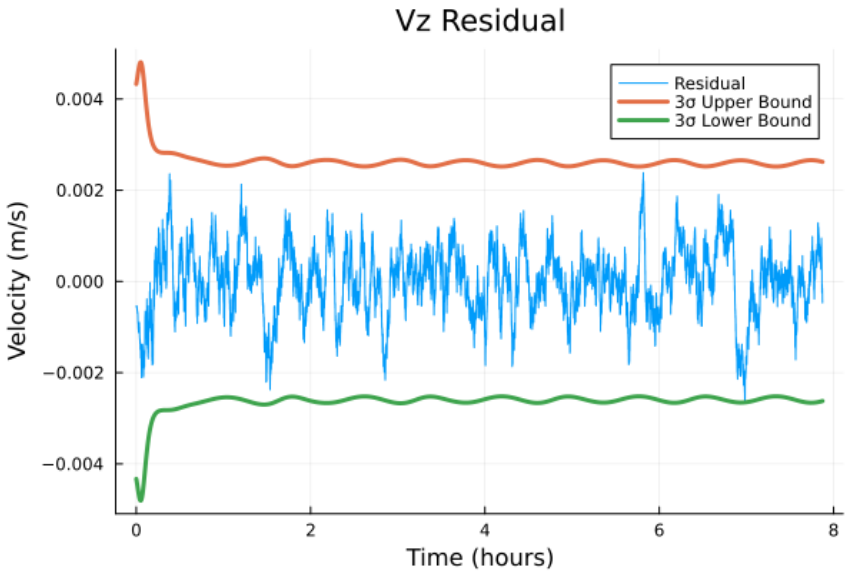


Figure 2.15: Estimator with DMC via random walk and the same order gravity models Z velocity residual

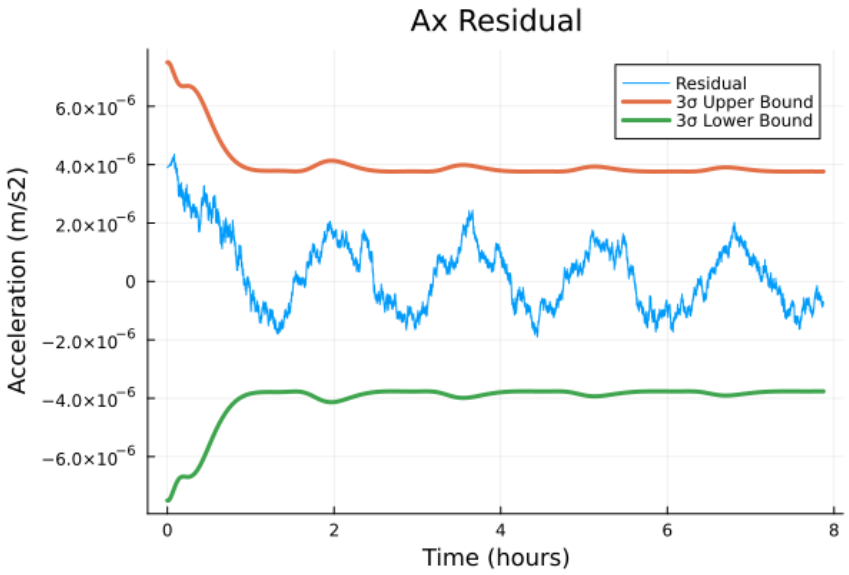


Figure 2.16: Estimator with DMC via random walk and the same order gravity models X unmodeled acceleration residual

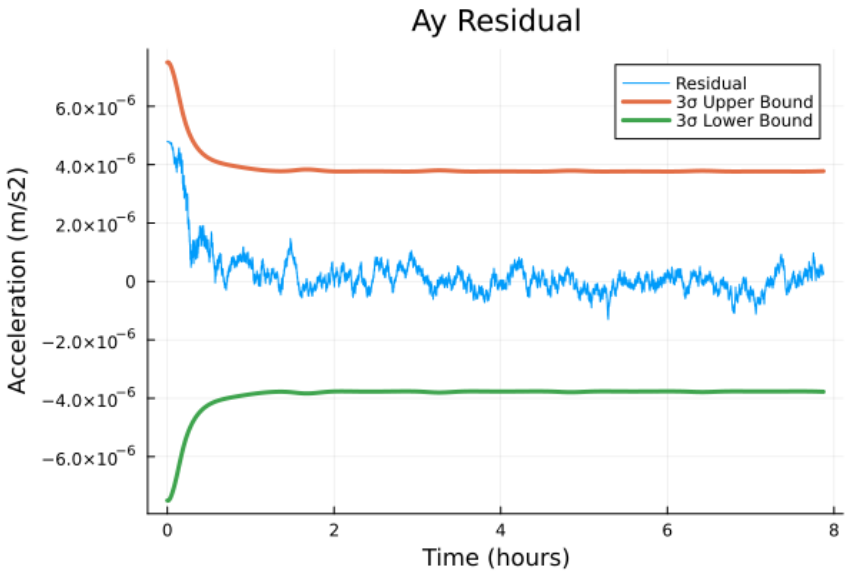


Figure 2.17: Estimator with DMC via random walk and the same order gravity models Y unmodeled acceleration residual

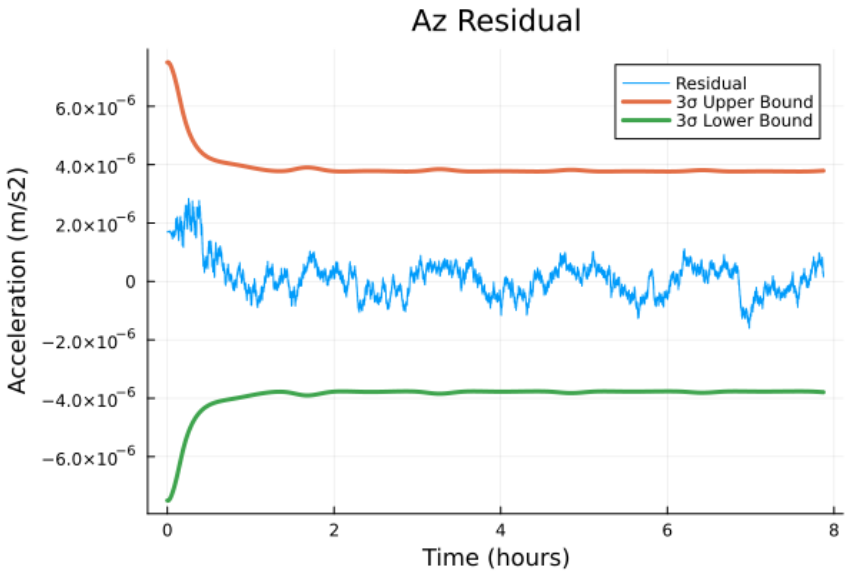


Figure 2.18: Estimator with DMC via random walk and the same order gravity models Z unmodeled acceleration residual

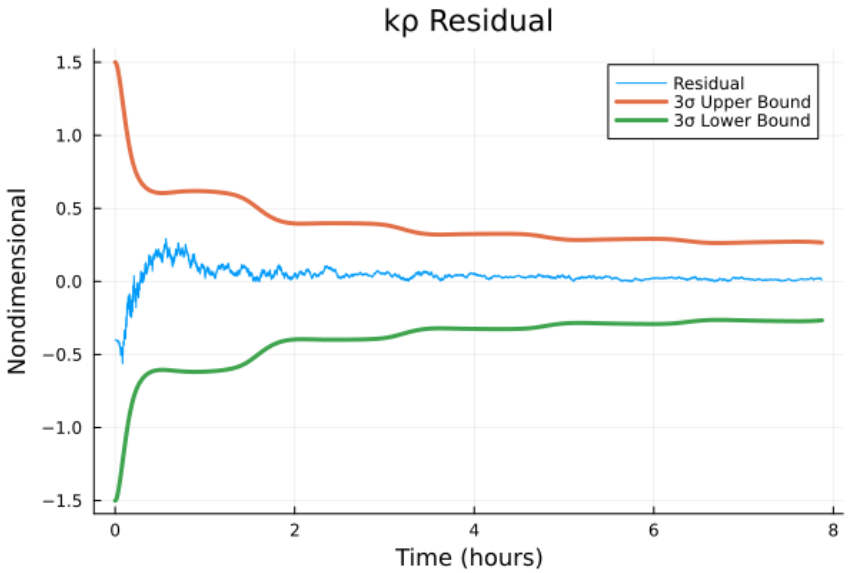


Figure 2.19: Estimator with DMC via random walk and the same order gravity models atmospheric scale residual

Estimator with DMC via random walk and different order gravity models

The unmodeled accelerations can be added to the state and estimated through the bias modeling techniques described in Section 2.3.4. This version estimates out the additional accelerations using a random walk model. The model assumes the dynamics of the unmodeled accelerations are governed by Gaussian noise and the scale of this noise determines the amount the user predicts the unmodeled accelerations to change. The process noise matrix for this version is shown in (2.34), and the residuals for the position and velocity are depicted in Figures 2.20, 2.21, 2.22, 2.23, 2.24, and 2.25. The residuals of the acceleration estimate shown in Figures 2.26, 2.27, 2.28 violate the three-sigma boundary at multiple time steps, therefore the filter is inconsistent for this part of the state. Modeling the unmodeled accelerations with a random walk is not the most optimal representation as a Gaussian does not capture the unmodeled acceleration dynamics well.

$$Q_{v_3} = \begin{bmatrix} (2.3094 \times 10^{-5})I(3) & 0 & 0 \\ 0 & (5.7735 \times 10^{-5})I(3) & 0 \\ 0 & 0 & (5 \times 10^{-6})I(3) \end{bmatrix}^2 \tag{2.34}$$

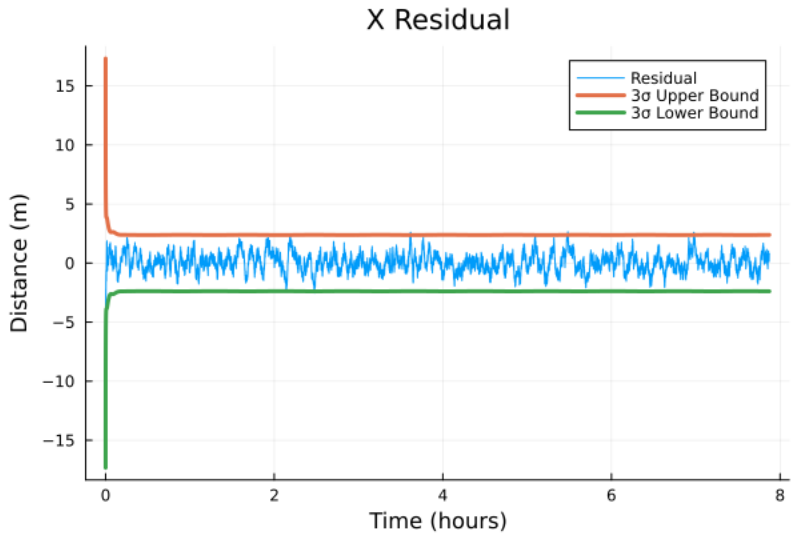


Figure 2.20: Estimator with DMC via random walk and different order gravity models X position residual

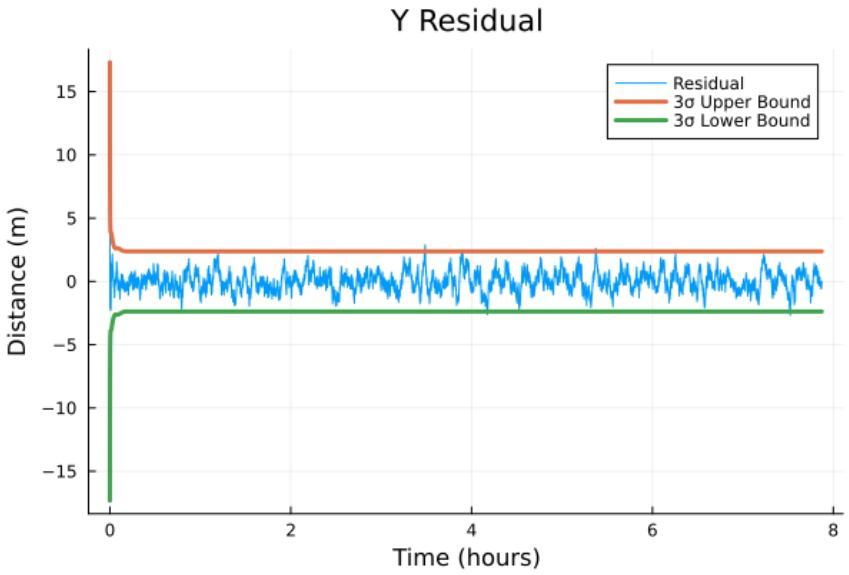


Figure 2.21: Estimator with DMC via random walk and different order gravity models Y position residual

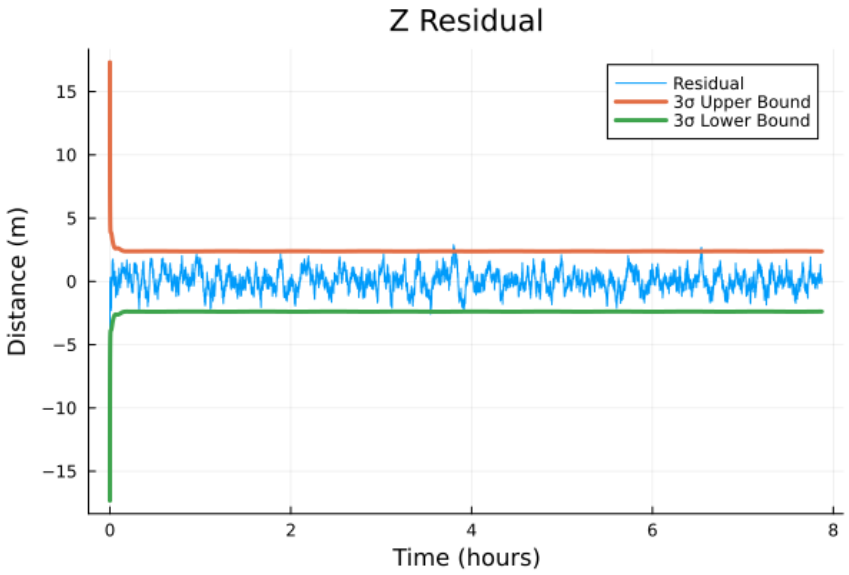


Figure 2.22: Estimator with DMC via random walk and different order gravity models Z position residual

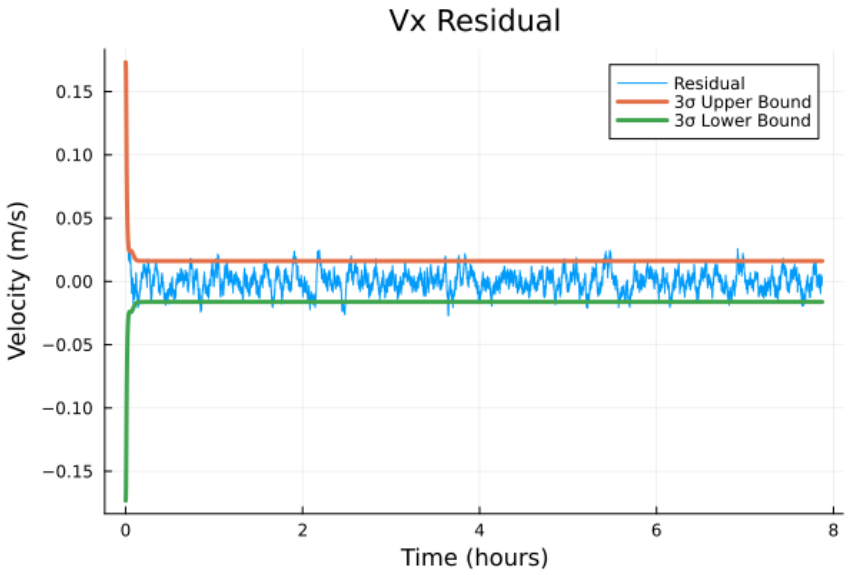


Figure 2.23: Estimator with DMC via random walk and different order gravity models X velocity residual

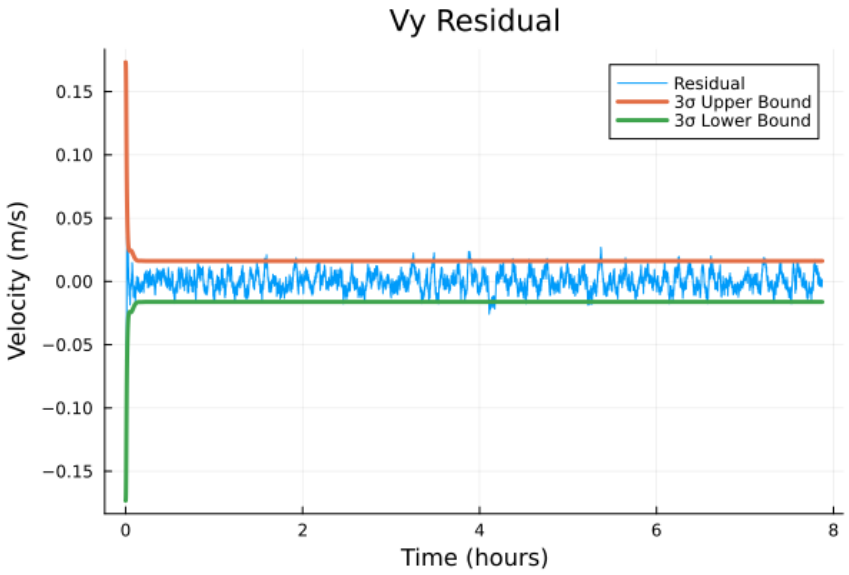


Figure 2.24: Estimator with DMC via random walk and different order gravity models Y velocity residual

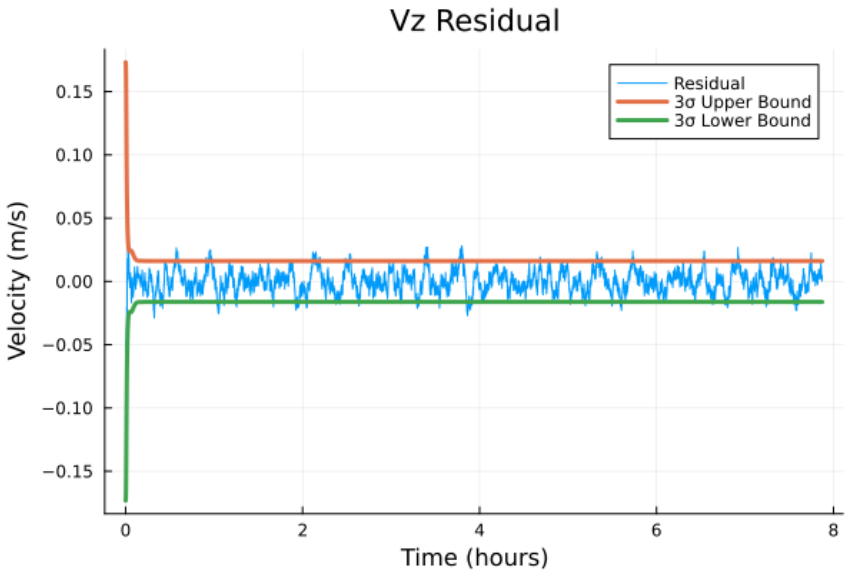


Figure 2.25: Estimator with DMC via random walk and different order gravity models Z velocity residual

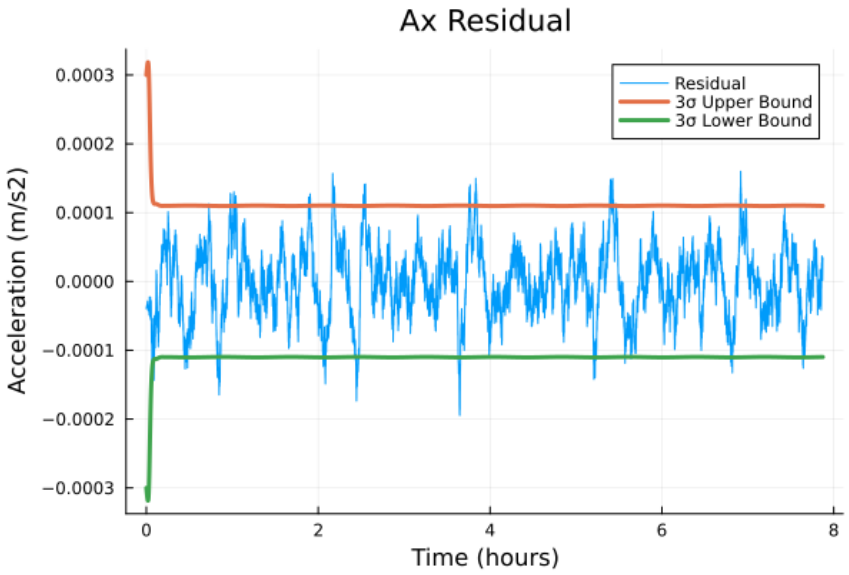


Figure 2.26: Estimator with DMC via random walk and different order gravity models X unmodeled acceleration residual

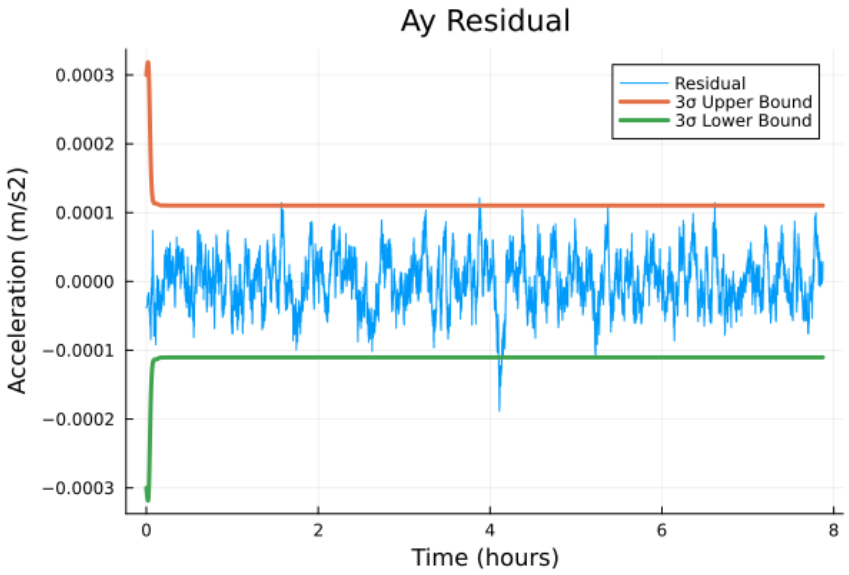


Figure 2.27: Estimator with DMC via random walk and different order gravity models Y unmodeled acceleration residual

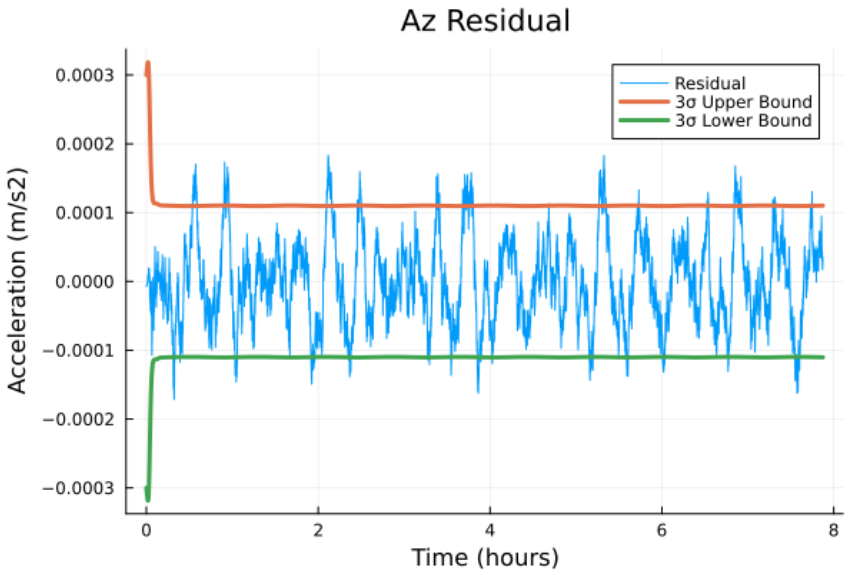


Figure 2.28: Estimator with DMC via random walk and different order gravity models Z unmodeled acceleration residual

Estimator with DMC via first-order Gauss-Markov model and different order gravity models

Another alternative to estimate the unmodeled accelerations is modeling their dynamics as a first-order Gauss-Markov stochastic process. This process is time correlated which is a more accurate representation unmodeled accelerations over time. An analytical representation of the process noise covariance is shown in (2.35) and derived in [17] by integrating the differential equations (2.24) and (2.23), and making approximations of the stochastic integrals.

$$
\begin{aligned}
Q_{k-1} &= \begin{bmatrix} \frac{h^4}{4}\Lambda & \frac{h^3}{2}\Lambda & \frac{h^2}{2}\Lambda & 0_{3\times3} \\ \frac{h^3}{2}\Lambda & h^2\Lambda & h\Lambda & 0_{3\times3} \\ \frac{h^2}{2}\Lambda & h\Lambda & \Lambda & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \Psi \end{bmatrix} \\
\Lambda &= \begin{bmatrix} q_{\varepsilon_x}[1-e^{-2\beta_x h}]/2\beta_x & 0 & 0 \\ 0 & q_{\varepsilon_y}[1-e^{-2\beta_y h}]/2\beta_y & 0 \\ 0 & 0 & q_{\varepsilon_z}[1-e^{-2\beta_z h}]/2\beta_z \end{bmatrix} \\
\Psi &= hQ_\beta
\end{aligned}
\tag{2.35}
$$

In (2.35), h is the simulation timestep which we set to 1 second and $q_{\varepsilon_{x,y,z}}$ are the diagonal elements of Q_ε. We assume the variances of each ε and β parameters are equal to σ_ε^2 and σ_β^2 respectively. We set $\sigma_\varepsilon^2 = 5.5e^{-11}$ and $\sigma_\beta^2 = 2e^{-9}$ to achieve consistent filter performance.

Overall, this adaptive method setting process noise covariance produced the best results with a position RMS error of 1.237 meters. However, this method is highly dependent on well-tuned noise values σ_ϵ and σ_β. These values were determined through inspection of the residual plots as well as the superimposed plot of the true unmodeled accelerations and the estimated unmodeled accelerations. A small value of σ_ϵ causes the estimate to quickly diverge and the residual plots to violate the covariance bounds. A large value of σ_ε causes large covariance bounds which permits larger errors in the estimate of the unmodeled accelerations. The position residuals are shown in Figures 2.29, 2.30, and 2.31, and the velocity residuals are displayed in Figures 2.32, 2.33, and 2.34. As seen acceleration residuals from Figures 2.35, 2.36,

and 2.37, this version of the filter is superior to version 3 in terms of consistency among all variables in the state as the residuals are bounded by the three-sigma covariance.

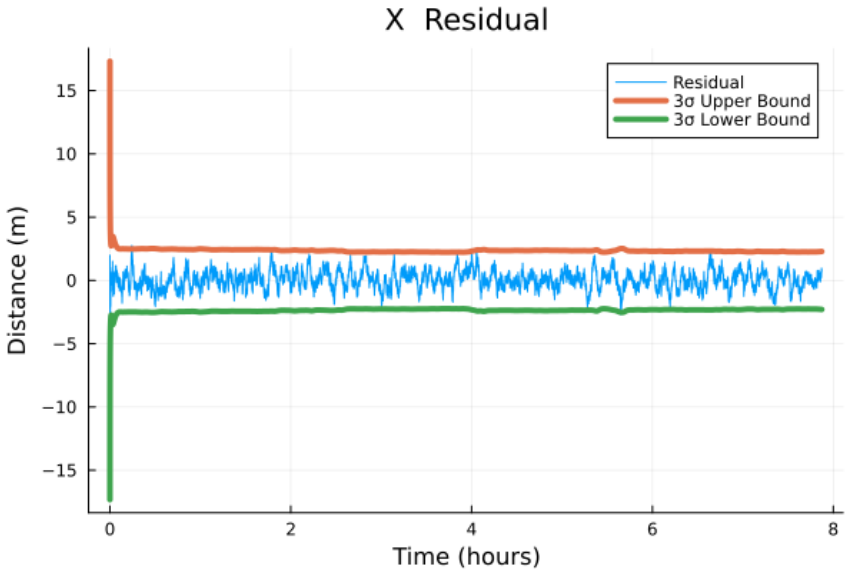


Figure 2.29: Estimator with DMC via first-order Gauss-Markov model and different order gravity models X position residual

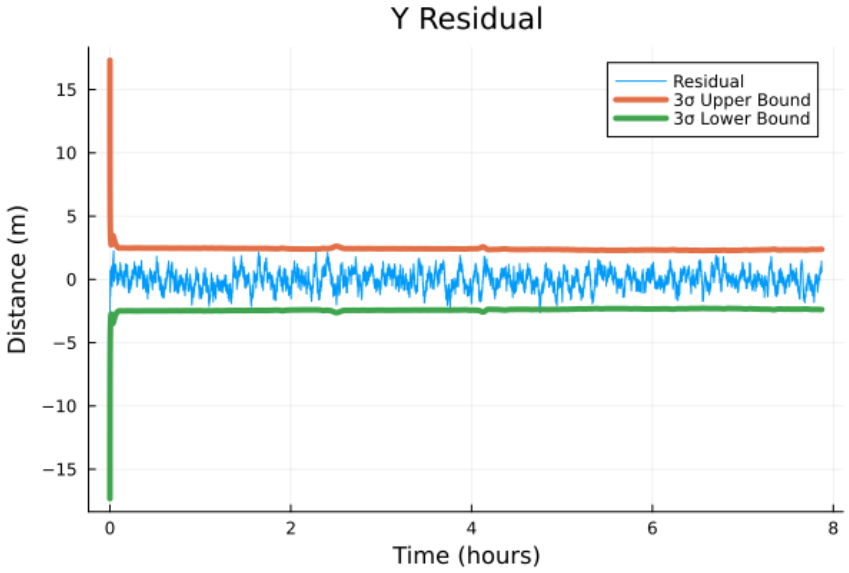


Figure 2.30: Estimator with DMC via first-order Gauss-Markov model and different order gravity models Y position residual

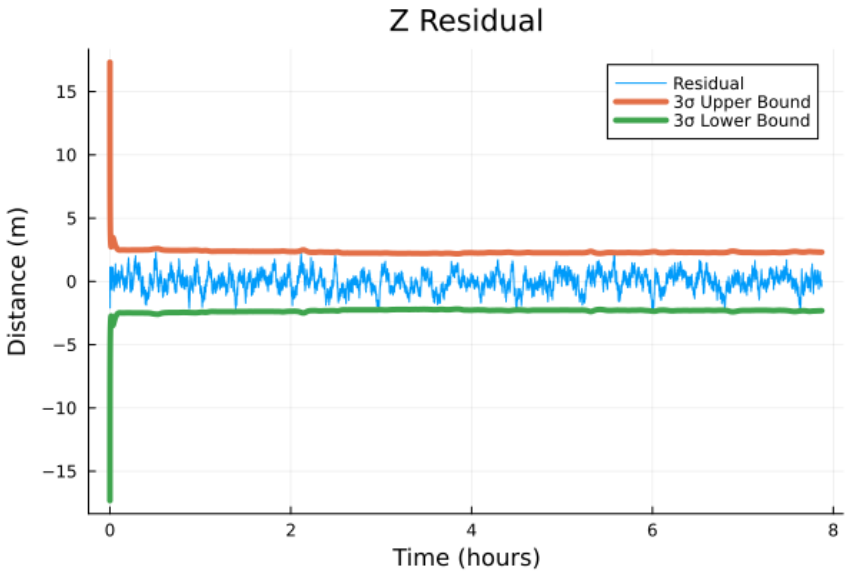


Figure 2.31: Estimator with DMC via first-order Gauss-Markov model and different order gravity models Z position residual

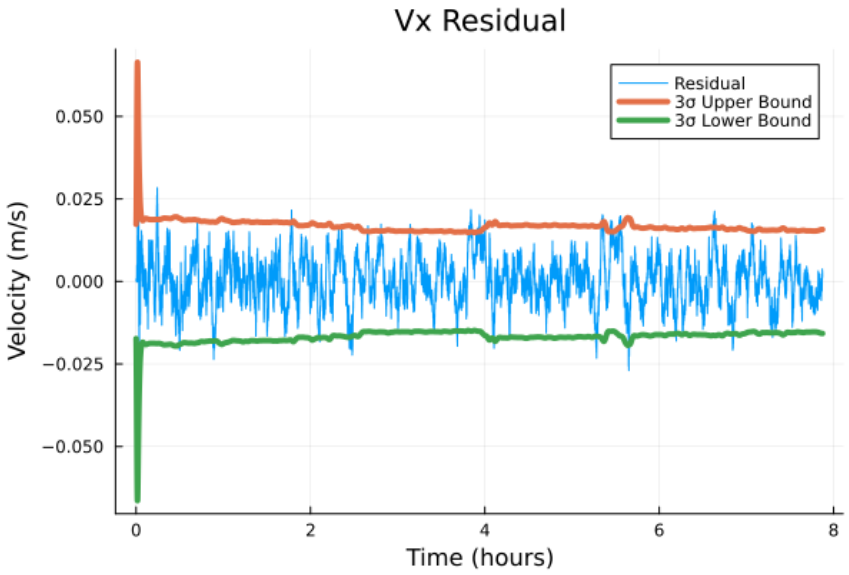


Figure 2.32: Estimator with DMC via first-order Gauss-Markov model and different order gravity models X velocity residual

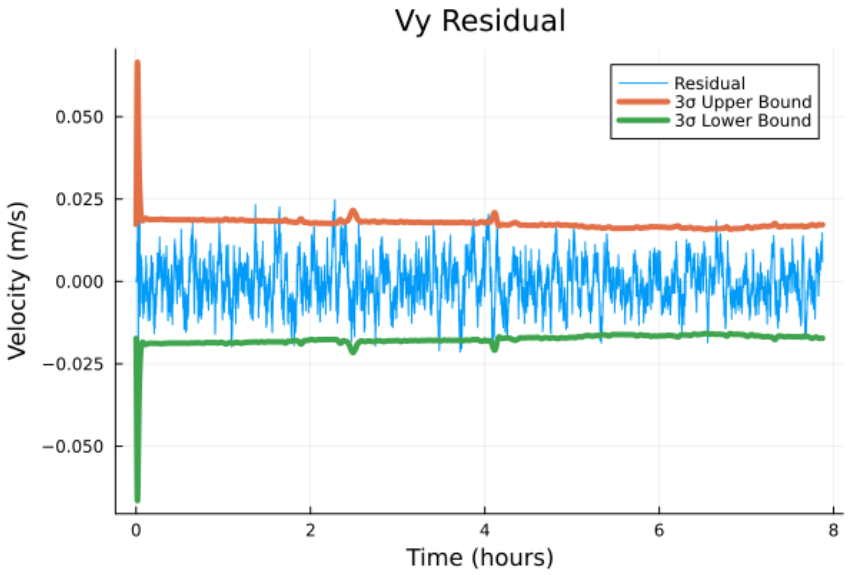


Figure 2.33: Estimator with DMC via first-order Gauss-Markov model and different order gravity models Y velocity residual

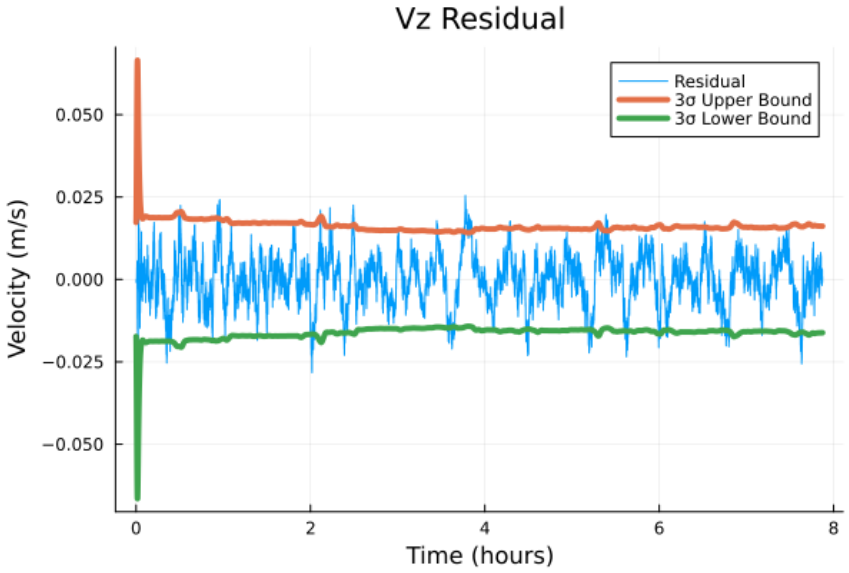


Figure 2.34: Estimator with DMC via first-order Gauss-Markov model and different order gravity models Z velocity residual

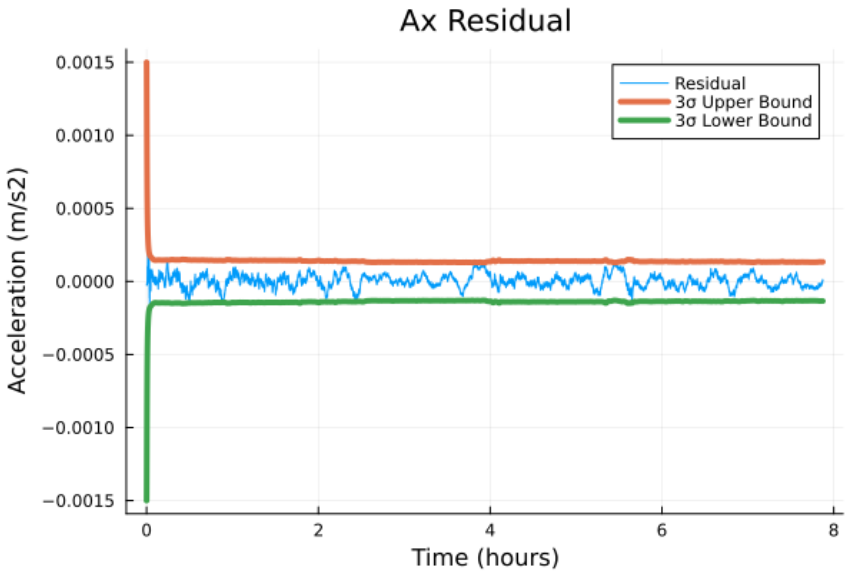


Figure 2.35: Estimator with DMC via first-order Gauss-Markov model and different order gravity models X unmodeled acceleration residual

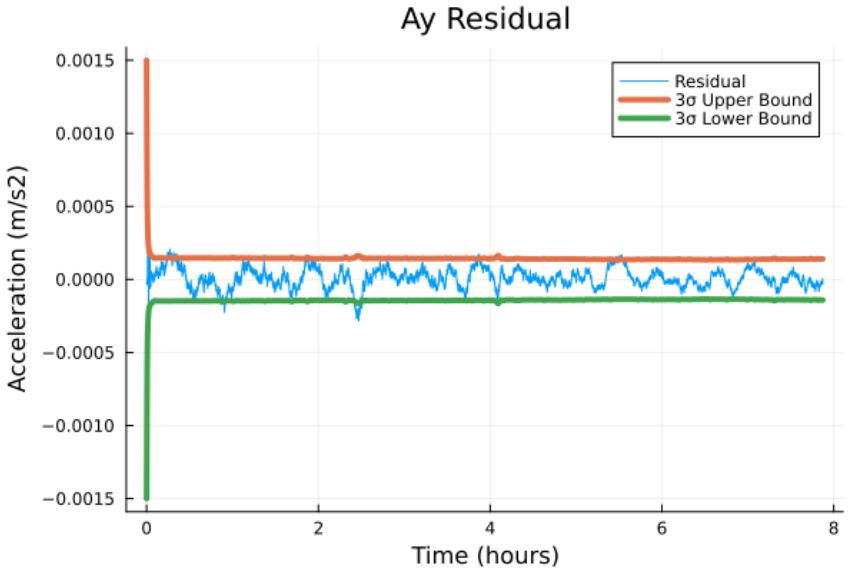


Figure 2.36: Estimator with DMC via first-order Gauss-Markov model and different order gravity models Y unmodeled acceleration residual

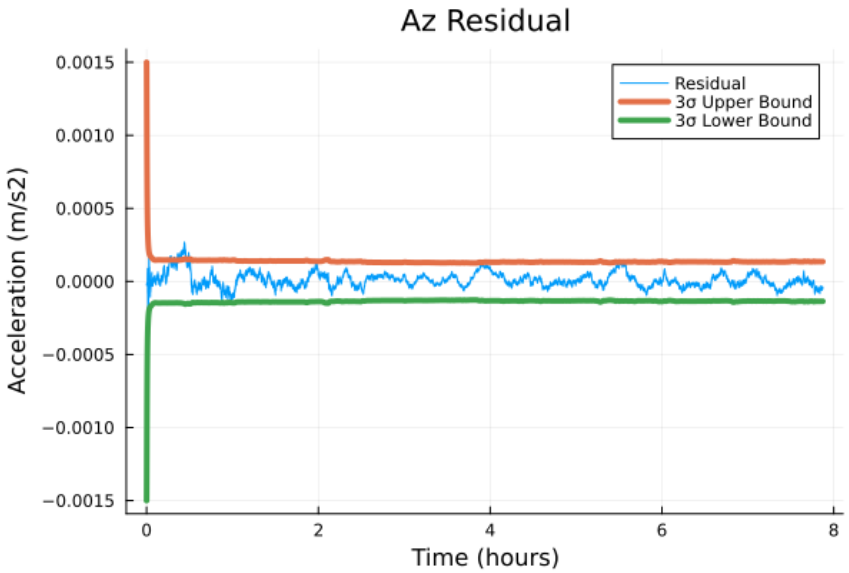


Figure 2.37: Estimator with DMC via first-order Gauss-Markov model and different order gravity models Z unmodeled acceleration residual

Chapter 3

Satellite-Based Wildlife Tracking and Simulation

3.1 Motivation

Existing methods for wildlife tracking include ARGOS and GPS, however these systems are costly and use large amounts of energy. We propose a new method using a constellation of CubeSats receiving signals from a lightweight transmitting tag weighing 1.4 g [18]. ARGOS claims to provide a tag position accuracy in the range of 150-2500 meters with a single satellite pass using Doppler measurements. Our approach consists a constellation of 4 satellites in LEO using time-of-arrival measurements to achieve accuracies under 100 meters. The lightweight tag will enable tracking of smaller animals to study migration patterns which can provide insight on natural resource exploitation and climate change patterns. Conservation efforts for endangered species is another benefit from accurate tracking data. ARGOS has a low sampling frequency which allows the tag to last for longer periods of time, however it provides noisy time series recordings. On the contrary, GPS has a high-sampling frequency with uncertainties lower than 50 m, but the receivers have short life expectancy [10]. Our method addresses these issues by using a low sampling frequency resulting in a long tag life along with a CubeSat constellation processing time-of-arrival measurements to provide accurate tracking.

3.2 Background

3.2.1 Measurement Models

Time-of-Arrival

The measurement for the time-of-arrival model is the transit time of the signal from the transmitting tag to the receiver onboard the satellite. We assume the initial transmission time is known and that the receive time is the same as the total transit time. The satellite and transmitter clocks are not synchronized which introduces a bias in the measurement. This bias is also estimated out in the algorithm as it can introduce errors of several meters in the estimation. The time-of-arrival measurement model is described in Equation 3.1:

$$ct_m = ||r_{tag} - Ar_{sat}|| + c\tau + I_\rho + \eta \tag{3.1}$$

$$A = \begin{bmatrix} cos\omega_E t_m & sin\omega_E t_m & 0 \\ -sin\omega_E t_m & cos\omega_E t_m & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

GPS literature refer to the product of the transit time (t_m) and the speed of light (c) as the pseudorange (ρ) as the result is a biased range due to additional additive error terms. However, we express the equation in terms of the transit time because this is the measurement we receive at every time step. The transit time (t_m) is a function of the true range added with a clock bias term (τ), an error term I_ρ that encapsulates the delay caused from the ionosphere, and random noise due to measurement uncertainty (η). The ionospheric delay is described in Section 3.5. In Equation 3.1, c represents the speed of light in a vacuum, and A is defined in Equation 3.2. This variable represents a rotation matrix to transform the position of the satellites (r_{sat}) from the Earth-Centered Inertial (ECI) frame to the Earth Centered Earth Fixed (ECEF) frame at the receiving time t. This transformation considers the rotation of the earth (ω_E) that happens as the signal is traveling. The earth rotation angle is small but can introduce an error of 10-20 m in the range computation which induces an east-west error in the position estimate [25]. Using

this model, we estimate the position of the tag (r_{tag}) in Cartesian coordinates in the ECEF frame as well as the clock offset term (τ).

Doppler Shift

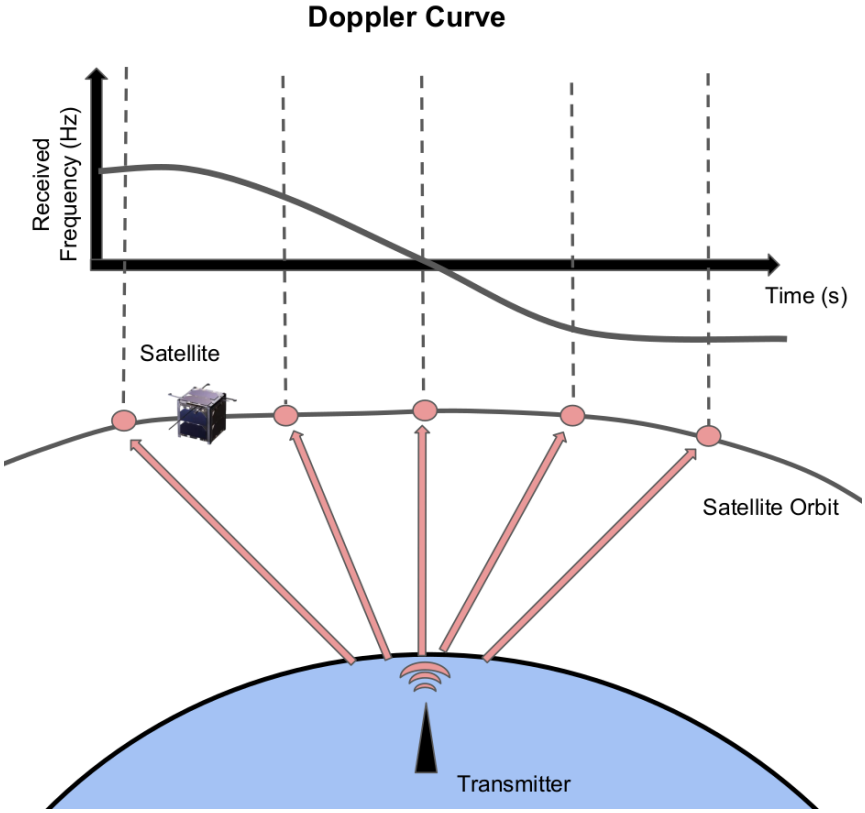


Figure 3.1: Behavior of Doppler curve as the satellite passes over the transmitter horizon. The y axis represents the difference between the received frequency and the original transmission frequency.

The Doppler measurement model records the change between the transmitted and received frequencies (Doppler shift) that occurs due to the relative motion of the spacecraft with respect to the tag. As the satellite approaches the tag, the received frequency is greater than the transmission frequency; and as the satellite moves away from the transmitter, the received frequency is lower than the transmission frequency. This effect is depicted in Figure 3.1. The Doppler measurement equation is shown in Equation (3.3), and it depends on the time derivative of the true range ($\dot{r}$), the

speed of light (c), the time derivative of the ionospheric effect ($\dot{I}$), the transmitted frequency (f_0), and measurement errors (η). Similar to the previous measurement model, the position of the tag (r_{tag}) is estimated in Cartesian coordinates as well as a frequency offset term ($\dot{b}$).

$$\Delta f = f_0 \frac{\dot{r} + \dot{I}}{c} + \dot{b} + \eta \tag{3.3}$$

The derivation of $\dot{r}$ and $\dot{I}$ are shown in the Appendix.

3.2.2 Newton's Method

Newton's method is an iterative root-finding method for a nonlinear equation. This equation is known as the residual ($r(x) = 0$) where x is the vector of unknown variables which is architecture dependent for this application. The unknowns for each architecture will be described in Section 3.6. For both cases, the residual is the predicted measurement (m_p) subtracted by the actual measurement (m) as shown in Equation (3.4).

$$r(x) = m_p - m \tag{3.4}$$

This method requires an initial guess $\bar{x}$ and the main idea of the algorithm is to approximate the nonlinear residual function as a linear function through a first-order Taylor series expansion as shown in Equation (3.5). Next, we find the additive update Δx that evaluates the linear function to zero when applied to the initial guess. This is done by solving Equation (3.5) for Δx. It is recommended to use the backslash operator rather than an inverse for this solve as it is more robust due to better numerical properties.

$$\begin{aligned} r(\bar{x} + \Delta x) &= r(\bar{x}) + \frac{\partial r}{\partial x} \,|_{\bar{x}}\, \Delta x \\ x &= \bar{x} + \Delta x \end{aligned} \tag{3.5}$$

Then we solve Equation (3.5) to compute the additive update (Δx). Next, we apply an backtracking line search known as the Armijo rule to ensure that the step agrees with the linearization within some user defined tolerance (b). This prevents the step from overshooting the root as it controls the magnitude of the step length by

Algorithm 2 Newton's Method

1: **for** $i = 1, 2, \ldots, N$ **do**
2: $\alpha = 1$ ▷ Initialize the step length
3: **if** $r(\bar{x}) < t$ **then**
4: break ▷ Check the residual against some tolerance (t)
5: **end if**
6: $J = (\frac{\partial r}{\partial \bar{x}} \mid_{\bar{x}})$ ▷ Calculate the jacobian of r and evaluate at $\bar{x}$
7: $\Delta x = J \backslash (-r(\bar{x}))$ ▷ Solve for the iterative update Δx
8: **while** $r(\bar{x} + \Delta x) > r(\bar{x}) + b\alpha J \Delta x$ **do** ▷ Apply an Armijo Linesearch
9: $\alpha = c\alpha$ ▷ Find the new step size
10: **end while**
11: $\bar{x} = \bar{x} + \alpha \Delta x$ ▷ Apply the iterative update
12: **end for**=0

multiplying α by a scale factor (c) if the Armijo condition is met. [15]

Then the additive update (Δx) is applied to the initial guess each iteration. This process is repeated until the residual is below a user defined tolerance (t) or the algorithm reaches the maximum amount of iterations (N). Reaching the maximum iterations means the residual did not meet the user defined tolerance and tuning the line search parameter (b) and tolerance (t) are potential solutions. The root-finding algorithm is summarized in Algorithm 2.

3.3 Uncertainty Propagation

The goal of the simulation is to propagate the uncertainty of the satellite positions and the TOA/Doppler measurements onto the tag position estimate. This is done using two distinct methods which result to the same solution. Both methods are described in the following subsections.

3.3.1 Monte-Carlo Simulation

A Monte-Carlo simulation relies on repeated random sampling and statistical analysis to compute a desired result [26]. For this application, the desired result is the Gaussian probability distribution of the tag state which is considered a random variable characterized by mean μ_t and covariance Σ_t. The satellite state and measurements

are also modeled as random variables with the satellite mean corresponding to a reference orbit trajectory and the standard deviation proportional to the satellite uncertainty result from the Version 4 state estimator in Section 2.3.6. Measurement uncertainty is proportional to hardware specifications/requirements. The Monte-Carlo simulation runs N-iterations of the Newton algorithm to solve for N tag positions and biases. We then compute the mean $\bar{x}_{tag}$ and covariance $\Sigma_{x_{tag}}$ of the N tag state estimates using Equations (3.6) and (3.7). The variable x_i represents one of the N tag state estimates.

$$\bar{x}_{tag} = \sum_{i=1}^{N} x_i \tag{3.6}$$

$$\Sigma_{x_{tag}} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x}_{tag})(x_i - \bar{x}_{tag})^T \tag{3.7}$$

3.3.2 Linear Approximation

A covariance can also be propagated by approximating the nonlinear residual function as a linear function. Then, we use the jacobian of the linear function to propagate the uncertainty of the function inputs (satellite states and measurements) onto the function outputs (tag state). This jacobian is found using the implicit function theorem.

Implicit Function Theorem

An implicit function has the form of equation (3.8)

$$r(x, y) = 0 \tag{3.8}$$

where x are the known function inputs and y are the unknown function outputs. For this application, the implicit function is the residual function where the inputs are the satellite positions and measurements, and the unknown is the tag position along with the bias. To compute the jacobian (A) that relates a change of the outputs to a change in the inputs, we use the implicit function theorem which is denoted in (3.9).

$$
\begin{aligned}
r(x,y) &= \frac{\partial r}{\partial x}\Delta x + \frac{\partial r}{\partial y}\Delta y = 0 \\
\Delta y &= -(\frac{\partial r}{\partial y})^{-1}\frac{\partial r}{\partial x}\Delta x \\
A &= -(\frac{\partial r}{\partial y})^{-1}\frac{\partial r}{\partial x} \\
\Delta y &= A\Delta x
\end{aligned}
\tag{3.9}
$$

Covariance Propagation

This jacobian is then used to propagate the uncertainty of the inputs (x) to the outputs (y) by equation (3.10). The derivation for Equation (3.10) is shown in the Appendix. The uncertainty of the inputs are represented by a covariance matrix (P_x) in which the diagonals correspond to the variance of each individual input variable.

$$P_y = AP_xA^T \tag{3.10}$$

3.4 Constellation Design

The satellite constellation geometry has a significant effect on the transmitter tracking accuracy. We create two polar orbits with an altitude of 500 km and place two satellites on each orbit. On these orbits, we parameterize the constellation geometry with three variables which are the true anomaly between two satellites on the same orbit (θ_{sep}), the true anomaly difference between the first satellite on each orbit (referred to as the delta true anomaly $\Delta\theta_{sep}$), and the RAAN separation between the two orbits (ω_{sep}). These variables are shown in Figure 3.2.

A 2D example of position estimation using range measurements is shown in 3.3. This example shows two cases in which a satellite pair compute range measurements with uncertainty (ϵ) to localize a transmitter. The difference between both cases is the placement of the satellites, however, the region of uncertainty is greater in 3.3b. This example can also be extended to the 3D case where region of uncertainty becomes a volume. This work characterizes the region of uncertainty through a covariance matrix which are calculated using a Monte-Carlo simulation and a linear

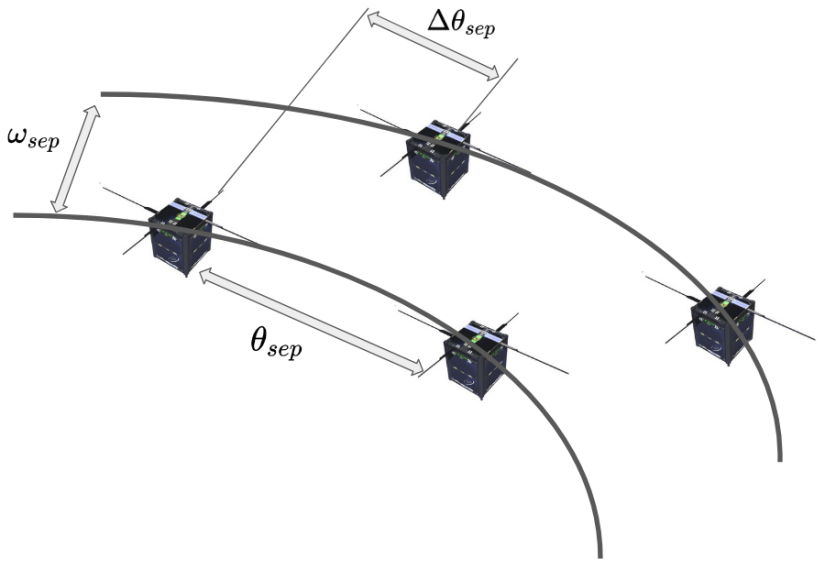
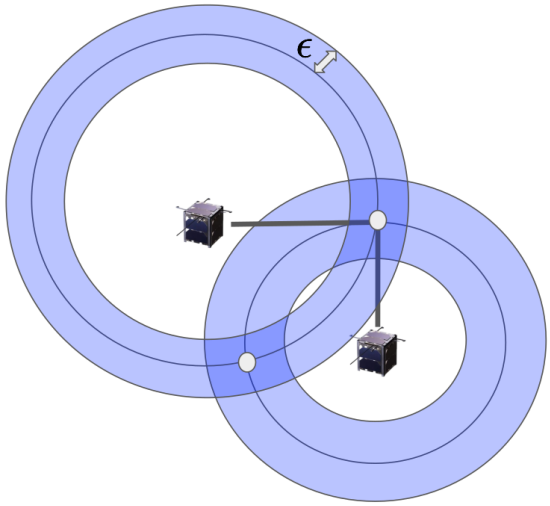
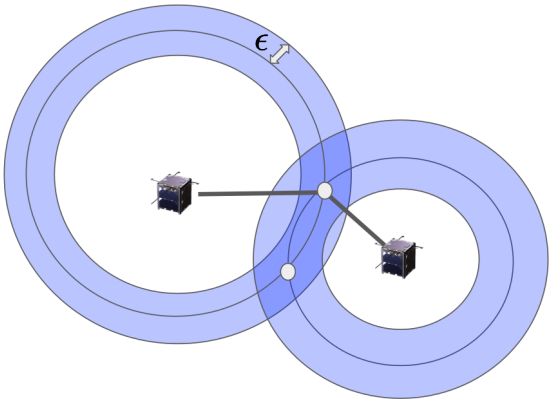


Figure 3.2: Satellite constellation showing the variables a user can modify to design the constellation.

approximation from Section 3.3.



(a) Transmitter position uncertainty based on satellite configuration 1



(b) Transmitter position uncertainty based on satellite configuration 2

Figure 3.3: Transmitter position uncertainty based on an alternate 2D satellite configuration. The gray points represent potential transmitter locations and the dark-shaded regions show the uncertainty of the tag position estimate

3.5 Additional Uncertainties

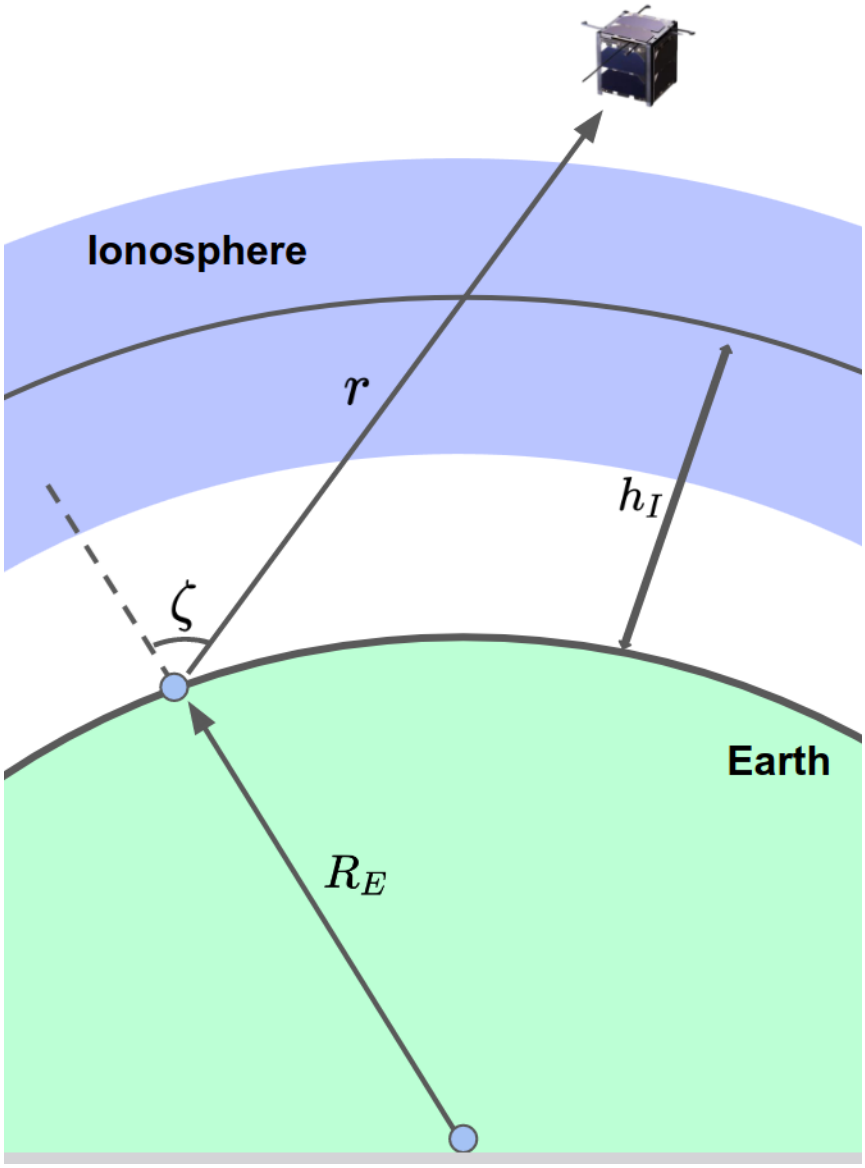


Figure 3.4: Ionosphere thin shell model used to determine the increased signal path length. The larger the zenith angle, the larger the delay

The ionosphere is a region of ionized gasses that is caused by the radiation from the sun, and it extend from about 50 km to 2000 km above the surface of the Earth

[27]. The physical characteristics of the ionosphere change widely between day and night, specifically the total electron count (TEC) in the path of the signal. The TEC is the number of electrons in a tube with a 1 m^2 cross section extending from the transmitter to the satellite. Models of the ionosphere such as the Klobuchar model and TEC maps exist [21], however uncertainty remains in the model as unexpected effects such as solar flares can drastically change the TEC value [25]. The ionosphere is modeled as a thin shell at a set height above the Earth (h_E). Equation 3.11 depicts the effect the ionosphere has on the pseudorange measurement. It is dependent on the zenith angle of the satellite (ζ), the radius of the Earth (R_E), and the frequency of the signal (f), and the TEC value in the vertical direction ($TECV$). The units of TEC are known as TECU and 1 TECU is equivalent to 1e16 electrons. A change by 1 TECU at the transmission frequency of 400 MHz causes a delay of 2.5 meters for the time-of-arrival method.

$$I(\zeta) = \frac{40.3TECV}{f^2}[1 - (\frac{R_E \sin\zeta}{R_E + h_I})^2]^{\frac{-1}{2}} \tag{3.11}$$

The effect of the ionosphere on the Doppler measurements is the time derivative of the ionosphere delay equation from the time-of-arrival method and is shown in equation 3.12 [3]. We derive this expression in the Appendix and compare it to the ionospheric correction equation in [20] to validate our expression. The zenith angle is a function of time, and the time derivative of the zenith angle was computed using the finite difference at neighboring time steps.

$$\dot{I}(\zeta) = \frac{(40.3TECV)(\frac{(R_E \sin\zeta)(R_E(\cos\zeta)\dot{\zeta})}{(R_E+h_I)^2})}{f^2(1 - (\frac{R_E \sin\zeta}{R_E+h_I})^2)^{\frac{3}{2}}} \tag{3.12}$$

3.6 Candidate System Architectures

Table 3.1 summarizes the system architectures that are considered along with a number label to reference these architectures in Figures 3.5 and 3.6.

Both single and dual-frequency cases were tested as conventional GPS utilizes a dual-frequency approach to mitigate the ionospheric error. The same result is observed in this simulation as the additional frequency provides more measurements

Label	Configuration
1	4 Satellites TOA 1 frequency
2	4 Satellites TOA 2 frequencies
3	4 Satellites Doppler 1 frequency
4	4 Satellites Doppler 2 frequencies
5	7 Satellites Doppler 1 frequency
6	7 Satellites Doppler 2 frequencies

Table 3.1: Candidate System Architectures

for the problem which allows the algorithm to estimate the TEC values for each path. Different variables were estimated out for each architecture, and these selections are summarized in Table 3.2.

Label	Tag Position	Tag Velocity	Clock Bias	Frequency Bias	TEC
1	X		X		
2	X		X		X
3	X			X	
4	X			X	X
5	X	X		X	
6	X	X		X	X

Table 3.2: Estimated parameters for each architecture

The advantages of the time-of-arrival (TOA) measurement model is the low measurement noise in the simulation as we assume the receiver onboard the satellites to have a timing accuracy of 20 ns, based on timing error tests conducted in [19]. The numerical conditioning of the TOA measurement equation is also superior to Doppler as the algorithm is able to converge in a few iterations and robust to several initial guesses. Timing accuracy is crucial for this method to work, as a timing error of 1 ms translates to a position error of 300 km. The use of one frequency obtains a biased tag position along with the clock bias. By using two frequencies, we estimate out the TEC values as these values have a high uncertainty due to the variability in the physical characteristics of the ionosphere and this yields better tag positioning accuracy and a lower bias.

The Doppler-based formations do not require precise time synchronization, but

the frequencies are required to be stable for long periods of time [25]. Frequency instability contributes significantly to tag positioning error. We assume an error of around 1 Hz for the frequency measurement due to the short transmission time for the proposed tag [3]. Another disadvantage of Doppler-based positioning is the weak information content regarding the tag position in the Doppler measurement equation as a change in the tag position does not significantly change the measurement. Difficulties converging also arise with the initial guess used in the algorithm. Doppler positioning also produces a biased answer when the tag has a moderate velocity. We assume the tag has no velocity in the algorithm, however the tag velocity changes the residual slightly to bias the estimate. The 4 satellite single frequency Doppler system estimates the tag position along with a frequency bias term, and the 7 satellite single frequency Doppler system approximates the same parameters along with the velocity of the tag. All the dual-frequency architectures additionally estimate out the TEC values for each satellite. For all the single frequency architectures, the TEC was treated as a random variable sampled from a scaled Gaussian distribution.

3.7 Architecture Accuracy Results

We implemented both a Monte-Carlo simulation and a linear approximation for uncertainty propagation to ensure the results of each coincide. First we set the tag position in geocentric coordinates to a longitude of -165.45, longitude of 0, and altitude of 0. Using SatelliteDynamics.jl, we convert to Earth centered Earth fixed (ECEF) coordinates and generate time-of-arrival and Doppler measurements using Newtons method with the known tag and satellite positions without uncertainty. We then formulate the residual from (3.4) and solve for the tag position and bias using Newton's method for each Monte-Carlo iteration. In each iteration, we include the satellite state and measurement uncertainties. We simulate 1000 Monte-Carlo runs and calculate the mean and covariance of the tag state. We repeat this process for every satellite position in the horizon of the tag. The horizon is set to 70 degrees from the satellite normal on both sides. For the TOA cases, the initial guess was the centroid of the satellite constellation projected onto the surface of the Earth. The algorithm converged in 6 iterations with tolerance set appropriately for each case. However, this initial guess did not result in convergence for the Doppler residual,

therefore an initial guess 25-50 km kilometers away in each dimension from the ground truth tag position resulted in convergence in 5 iterations.

Figures 3.5 and 3.6 show the accuracy and bias of each architecture with respect to a user defined base case. This base case is defined in Table 3.3, and was determined to comply with mission constraints. We selected an altitude of 500 km to ensure we follow NASA orbital debris mitigation requirements which assures that the CubeSat won't pose an unacceptable hazard to other orbiting spacecraft, will deorbit in a reasonable amount of time, and no large piece of the CubeSat survives reentry when it deorbits and burns up in the atmosphere [1]. The RAAN, true anomaly, and delta true anomaly are proposed to change using drag modulation to control the relative positions of the satellites in a formation [12].

Base Case	
Altitude	500 km
RAAN separation	2°
True Anom. Sep.	10°
Delta True Anom. Sep.	3°
Time Accuracy	20 ns
Frequency Accuracy	1000 mHz

Table 3.3: Orbital parameters for the ground truth model

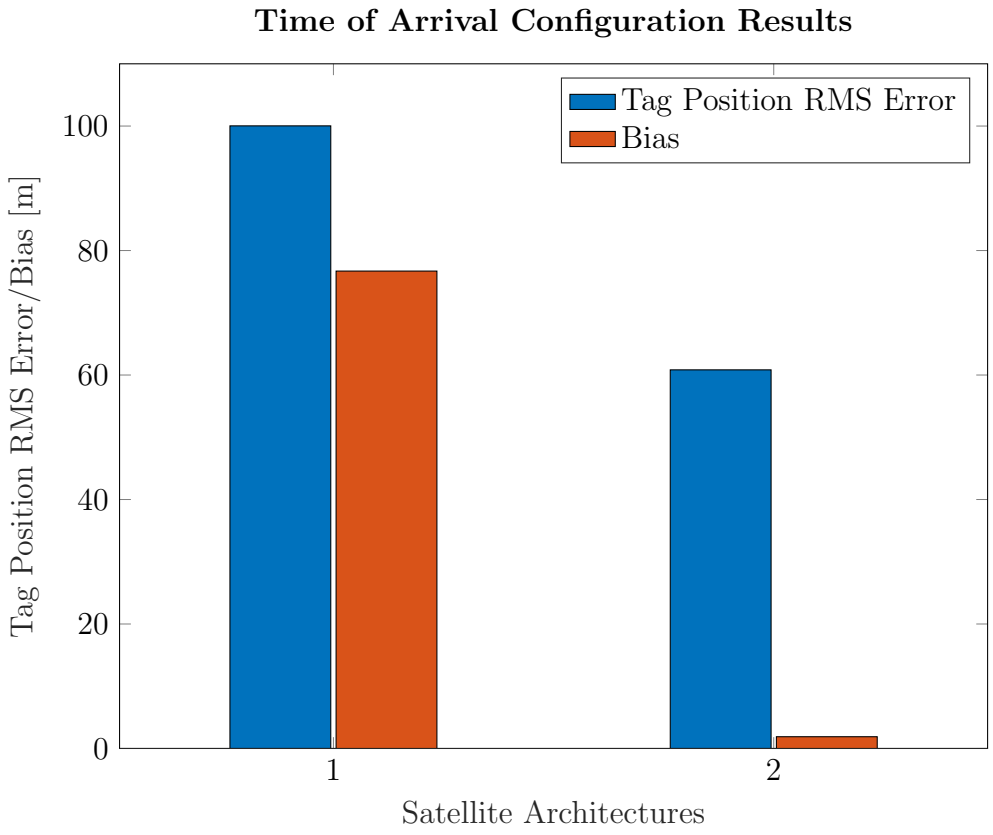


Figure 3.5: Accuracy of the tag position as the RAAN changes in the satellite configuration orbits. This analysis is done on the 4 satellite dual-frequency time-of-arrival case.

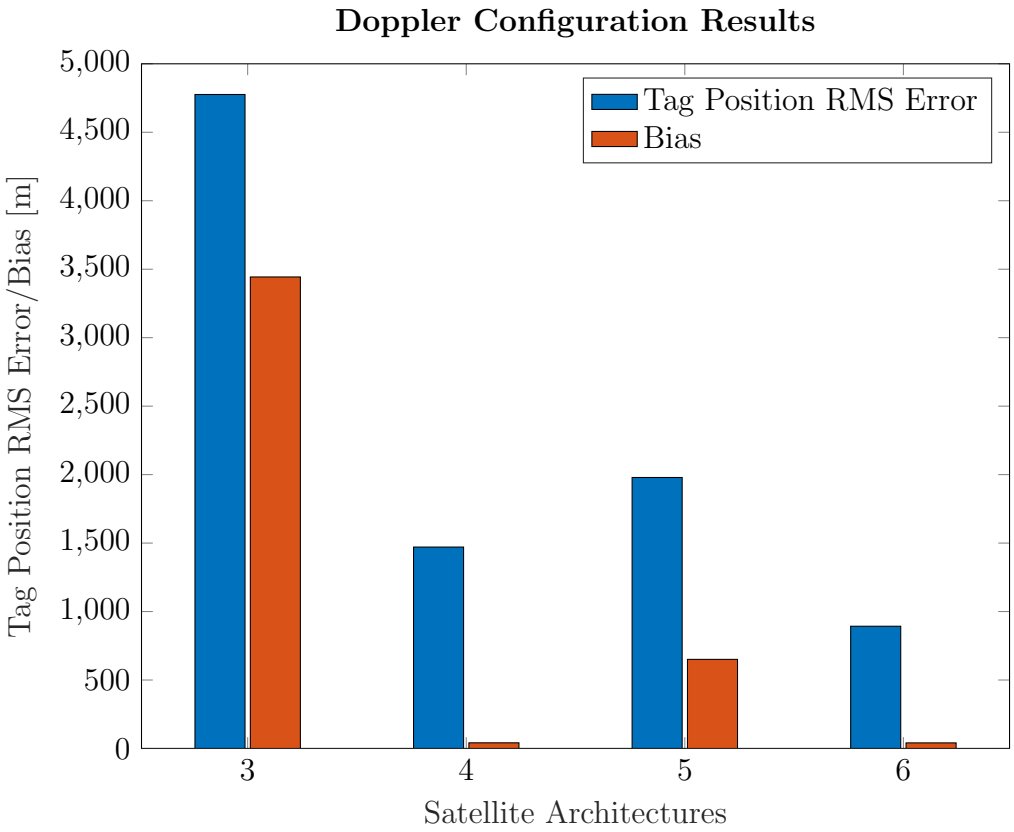


Figure 3.6: Accuracy of the tag position as the RAAN changes in the satellite configuration orbits. This analysis is done on the 4 satellite dual-frequency time-of-arrival case.

The dual-frequency architectures outperformed the single frequency cases for all the configurations which is expected as the ionospheric term is not treated as a random variable. Overall the 4 satellite dual-frequency case resulted in the best configuration with a tag position RMS error of 60.83 meters and a bias of 1.89 meters.

Next, we investigate the role of constellation geometry on the tag position RMS error. We conduct this analysis on the 4 satellite dual-frequency case since it provided the best results and use the base case from Table 3.3. We vary the RAAN separation, delta true anomaly, true anomaly, frequency separation, clock accuracy, and the tag transit time one at a time to analyze the tag position RMS error. Figures 3.7, 3.8, 3.9 show the results of the effect the satellite constellation geometry has on the tag position RMS error. These results show that an optimal configuration that minimizes

tag positioning accuracy consists of a RAAN separation of approximately 3 degrees, a true anomaly separation of 8 degrees, and a delta true anomaly separation as large as possible. Increasing the delta true anomaly reduces the amount of time all the satellites are in the horizon of the tag, therefore it becomes a trade off for the user to determine. Also, the higher the frequency separation, the lower the tracking accuracy as the difference in measurements is higher due to the difference in ionospheric effects, and this difference improves the conditioning of the problem. The clock accuracy plot in Figure 3.11 shows the position accuracy improves linearly as the clock accuracy improves. Similarly, we correlate the frequency measurement accuracy with the tag transmit time, and show that the longer the transit time results in a more accurate tag position RMS error.

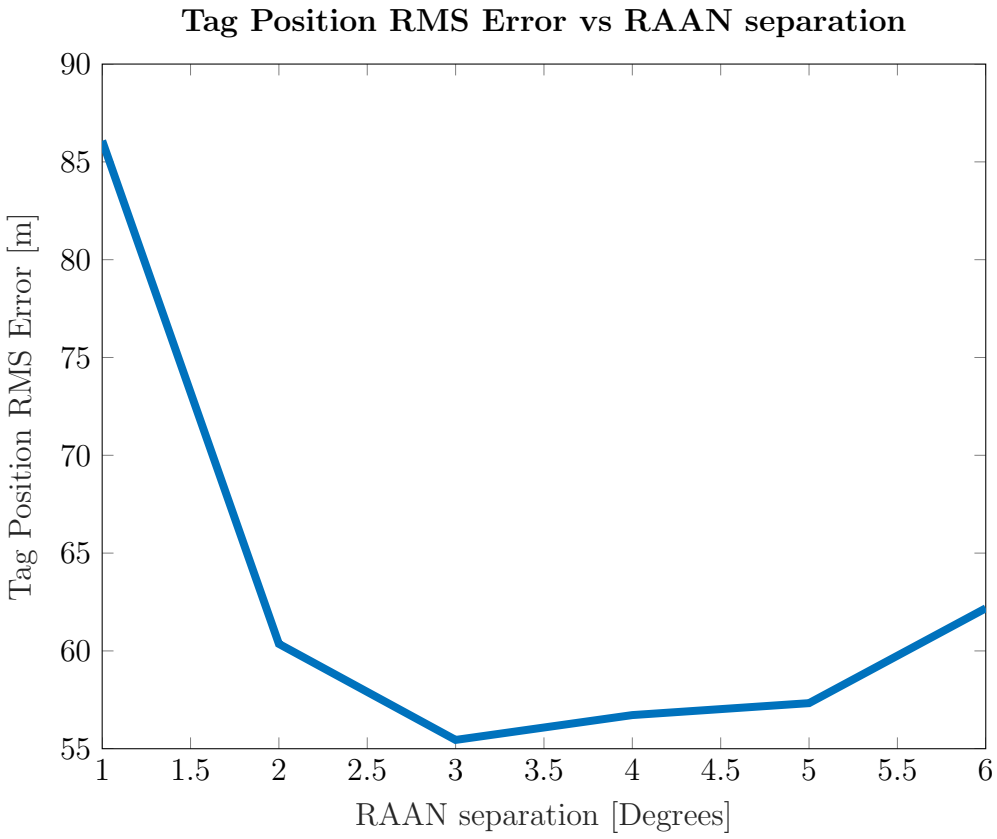


Figure 3.7: Accuracy of the tag position as the RAAN changes in the satellite configuration orbits. This analysis is done on the 4 satellite dual-frequency time-of-arrival case.

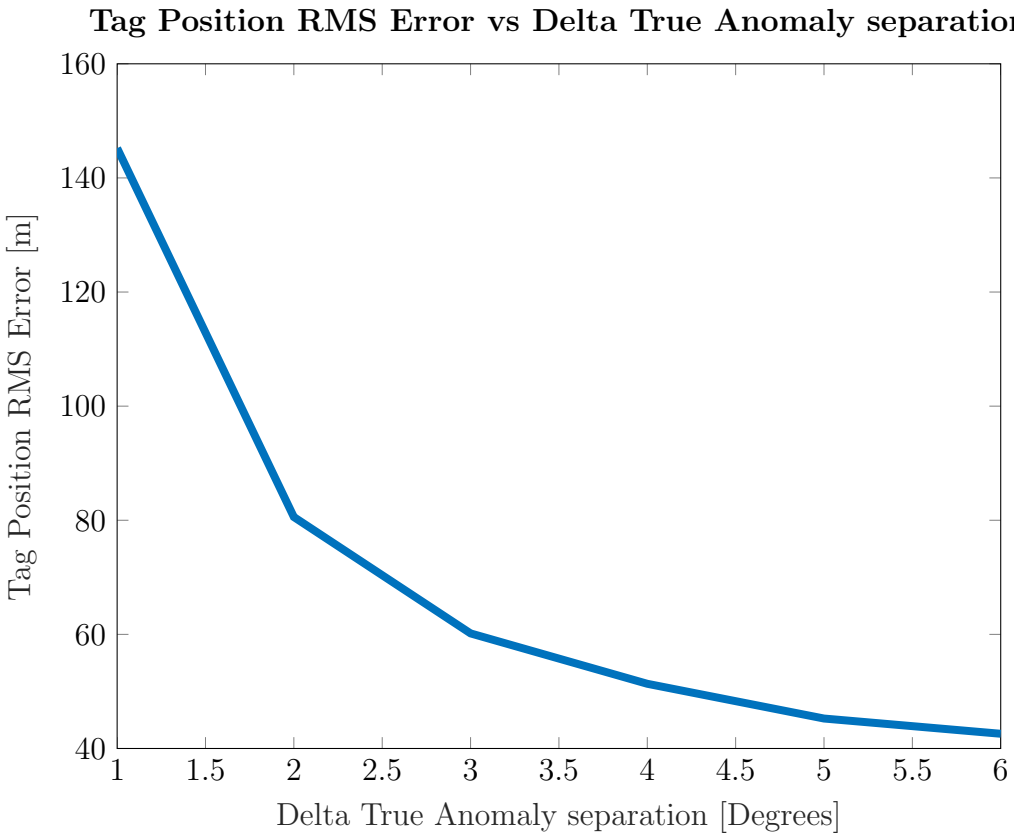


Figure 3.8: Accuracy of the tag position as the delta true anomaly changes in the satellite configuration. This analysis is done on the 4 satellite dual-frequency time-of-arrival case.

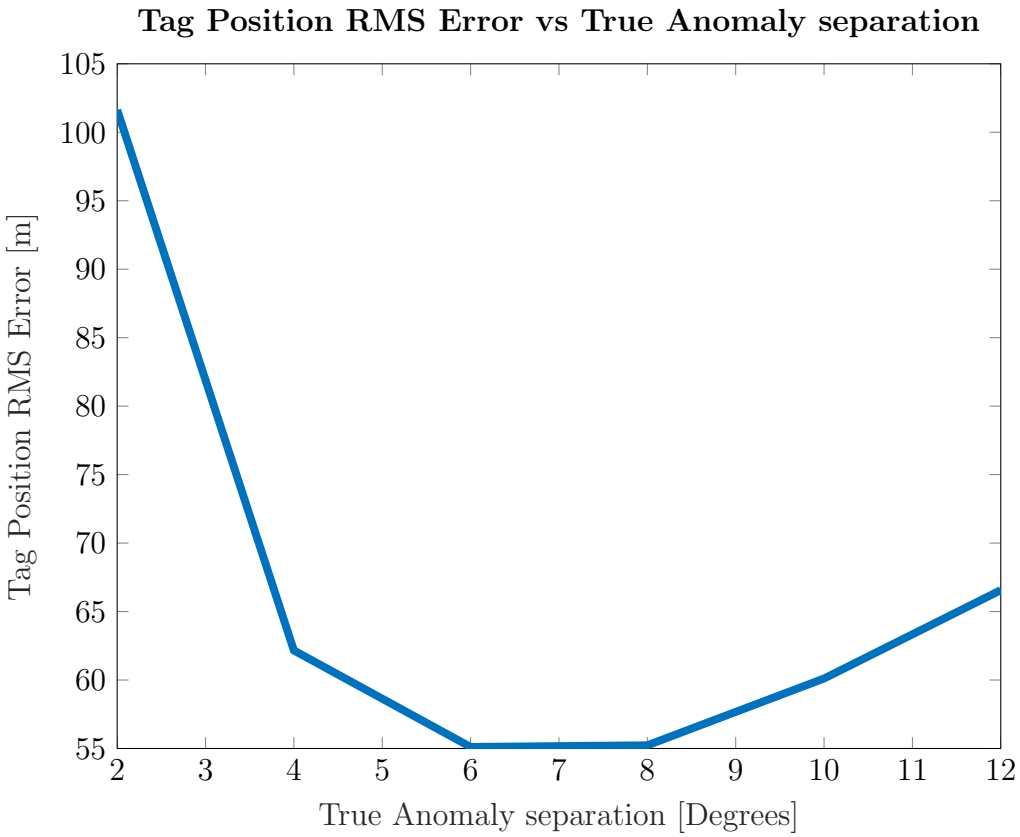


Figure 3.9: Accuracy of the tag position as the true anomaly between 2 satellites on the same orbit changes in the configuration. This analysis is done on the 4 satellite dual-frequency time-of-arrival case.

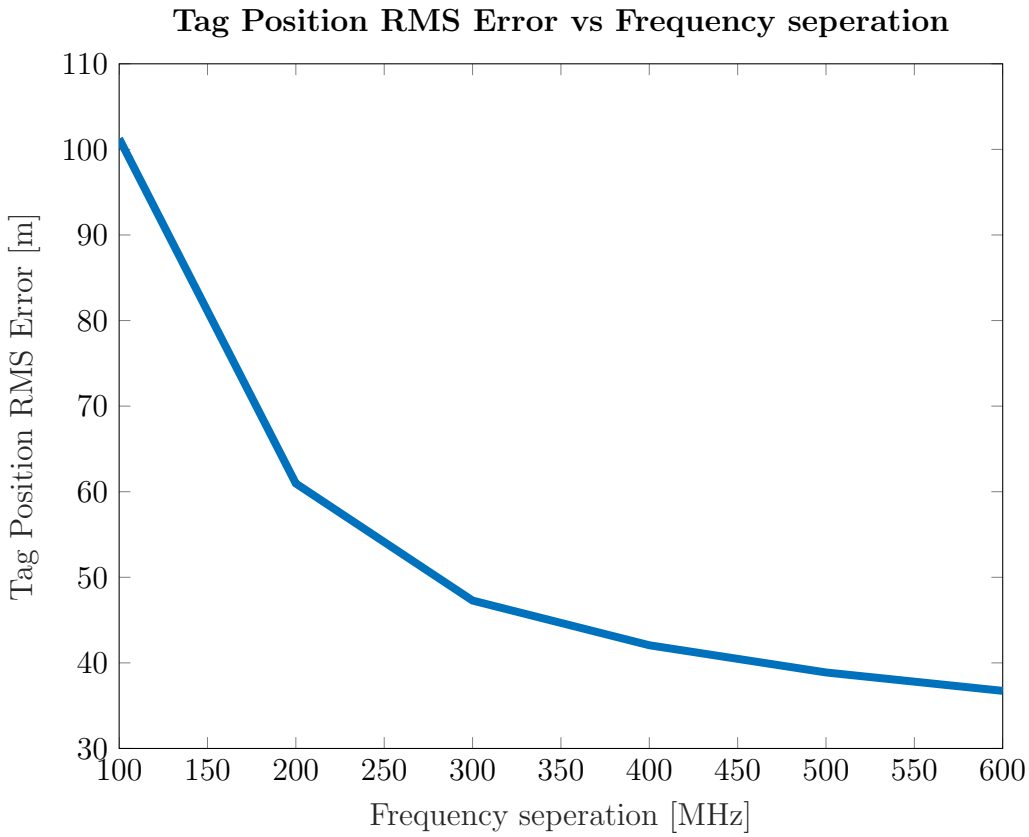


Figure 3.10: Accuracy of the tag position as the frequencies in the 4 satellite dual-frequency time-of-arrival case change. A base frequency of 400 MHz is used while the second frequency was incremented by intervals to generate this negative exponential behavior.

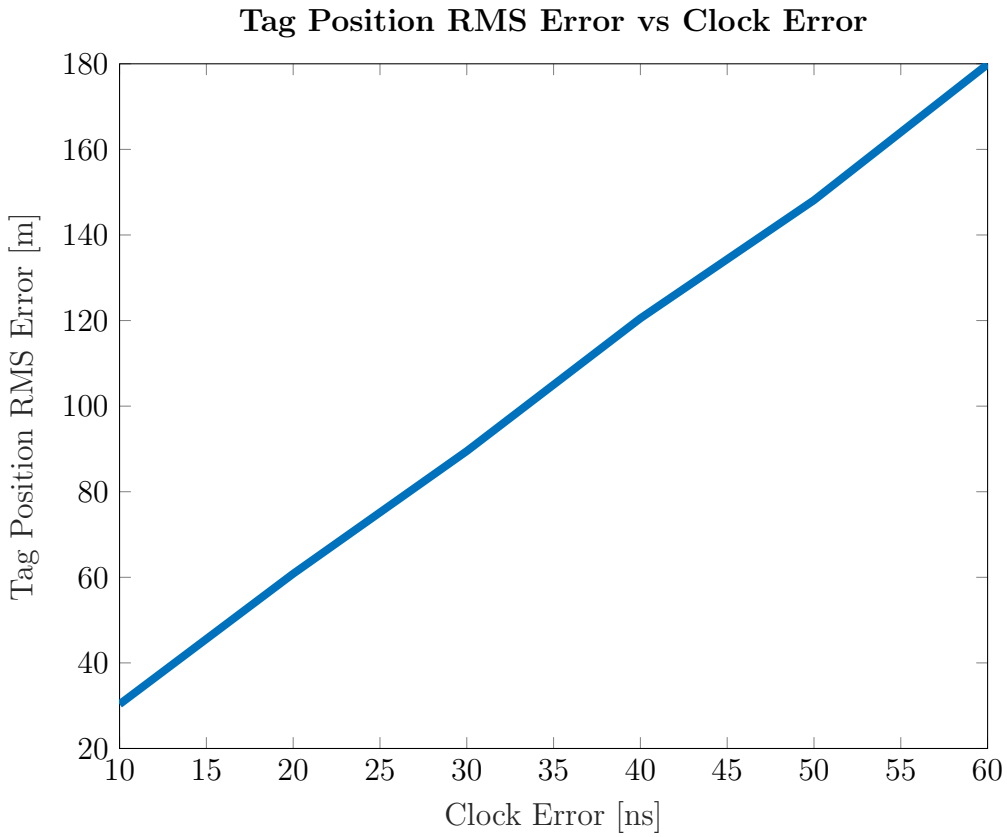


Figure 3.11: Accuracy of the tag position as the clock accuracy changes. This analysis is done on the 4 satellite dual-frequency time-of-arrival case.

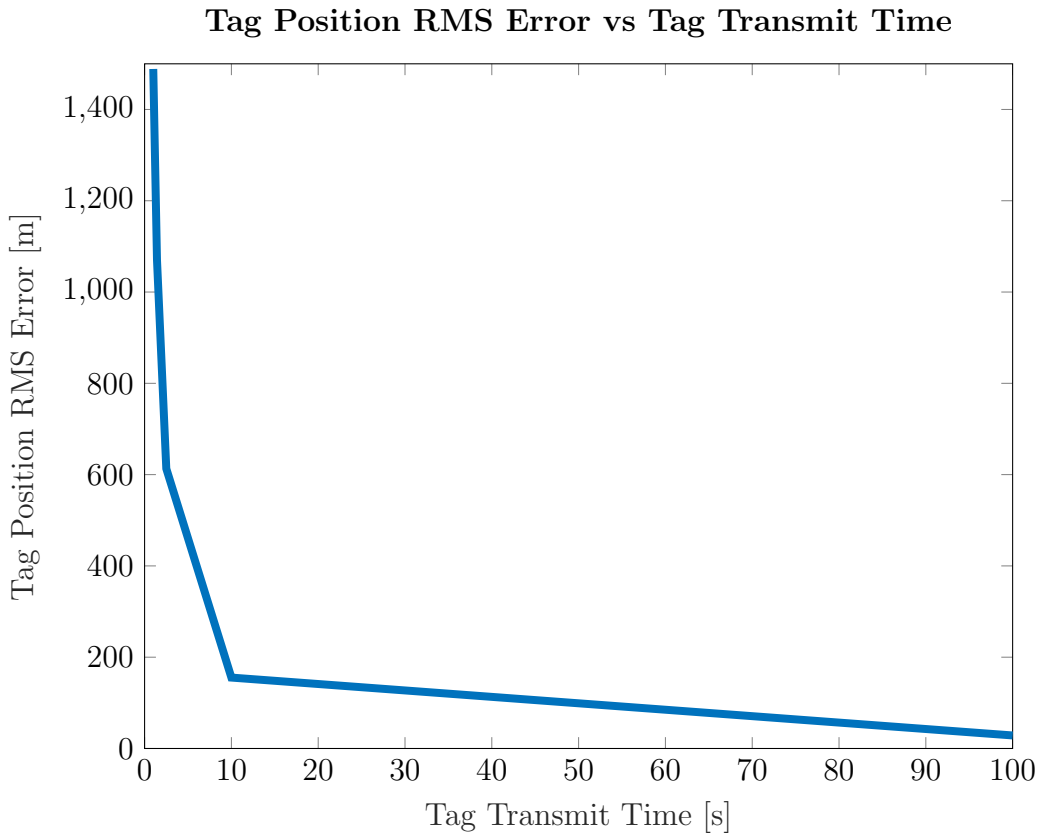


Figure 3.12: Accuracy of the tag position as the tag transmit time changes. This analysis is done on the 4 satellite dual-frequency Doppler case.

3.8 Interactive Simulation

We develop an interactive software modeling tool in Makie [8] to visualize the effects the satellite formation has on the tag position RMS error. This tool places a four satellite formation on two orbits (two satellites on each orbit), and allows the user to change orbital parameters of both orbits such as the orbit attitude, right ascension of the ascending node (RAAN) separation between both orbits, and the true anomaly separation between the satellites. The user can also vary the tag latitude to observe the tag position accuracy at different latitudes. The measurement noise (timing and frequency accuracy) can also be adjusted based off the hardware specifications of the proposed sensors onboard the satellite. The result of the tool is a covariance ellipse

in the north east up coordinate system to display the two-sigma confidence bound of the mean along with the position RMS error of both TOA and Doppler architectures. This software used the least squares approach to calculate the tag covariances since this method for finding the tag covariance is significantly faster than a Monte-Carlo simulation. Figure 3.13 shows the layout of the tool. The package is open-source and available in the following link:
https://github.com/vegaf1/SatelliteNavigation.jl

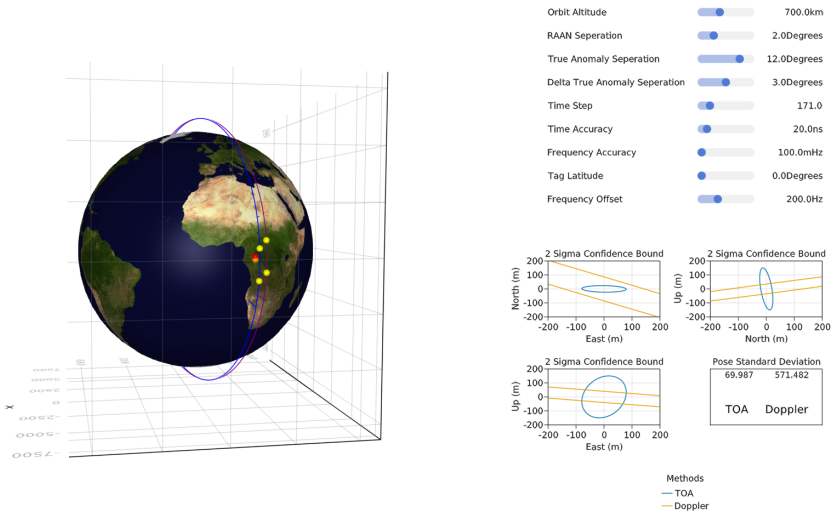


Figure 3.13: Layout of the interactive tool created to analyze the tag position accuracy using 2 frequency TOA and Doppler configurations.

Chapter 4

Conclusions

We present a simulation for satellite based tracking of a small transmitting tag using a constellation of cube satellites, along with an open source package to test user defined satellite geometries using the time of arrival and Doppler architectures. We also calculate the satellite state uncertainty using an EKF with bias modeling, and we include measurement errors and ionospheric effects in the simulation. The 4 satellite time of arrival dual frequency configuration is recommended due to the performance, complexity, and cost. This architecture had the best positioning accuracy and lowest bias due to the high timing accuracy for the receivers and estimating out the TEC terms in the ionosphere. This method also provides better accuracy than the current method used in wildlife tracking (ARGOS), and is more cost effective than both GPS and ARGOS.

Future work includes implementing a batch least squares solution of several orbits to initialize the satellite state estimation problem as this method optimizes a cost function rather than iteratively propagating a state and covariance, which may improve the sequential satellite state estimate. Also we plan on implementing a unscented Kalman filter (UKF) to compare it to the EKF as the UKF handles nonlinearities more efficiently due to propagating several points through the nonlinear functions rather than 1. Combining both time of arrival measurements and Doppler measurements is also interesting as more variables such as the tag velocity could be estimated out. We also plan to set up an optimization problem to obtain the most effective satellite constellation as there may be nonlinear relationships between the

constellation design parameters.

Appendix A

A.1 Earth Gravity Expansion to J2

This section will provide a derivation of the gravitational potential up to the J2 term.

$$U = \frac{GM_{\oplus}}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \frac{R_{\oplus}^{n}}{r^{n}} P_{nm}(sin\phi)(C_{nm}cos(m\lambda) + S_{nm}sin(m\lambda)) \tag{A.1}$$

For only J2, n=2 and m=0, which leads to the following simplification.

$$U = \frac{GM_{\oplus}}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \frac{R_{\oplus}^{n}}{r^{n}} P_{nm}(sin\phi)(C_{nm}) \tag{A.2}$$

$$U = \frac{\mu}{r}[P_{0,0}(sin(\phi)C_{0,0}) + (\frac{R_{\oplus}}{r})P_{1,0}(sin\phi)C_{1,0} + (\frac{R_{\oplus}}{r})^{2}P_{2,0}(sin\phi)C_{2,0}] \tag{A.3}$$

From the table on page 60 in [24], the geopotential coefficients are the following: $C_{0,0} = 1, C_{1,0} = 0, C_{2,0} = -1.08264e^{-3}$. The value of $C_{2,0}$ is the same value as variable J_2 in Equation (2.6) but negative. Next, we need the corresponding lower-order Legendre polynomials which are obtained on page 57 of [24]. The polynomials are the following: $P_{0,0} = 1$ and $P_{2,0} = \frac{3}{2}sin^{2}\phi - \frac{1}{2}$. Substituting these values in results in the gravity potential of Earth up to J2 which is shown in Equation A.4.

$$U = \frac{\mu}{r}(1 + \frac{J_2 R_\oplus^2}{2r^2}(1 - 3sin^2\phi)) \tag{A.4}$$

By converting Equation (A.4) to Cartesian coordinates and then taking the gradient through a symbolic differentiation package such as Symbolics.jl [13], one obtains Equation (2.6).

A.2 Doppler Measurement Equation Derivation

The Doppler measurement model is dependent on the time derivative of the pseudo range and the derivation for this term is shown below. Equation (A.5) derives the time derivative of the true range. The matrix A is the rotation of the Earth that occurs while the signal is being transmitted and transforms the ECI satellite position into the same ECEF frame as the tag.

$$\begin{aligned}
r &= \sqrt{(r_{tag} - Ar_{sat})^T(r_{tag} - Ar_{sat})} \\
r &= ((r_{tag}^T - r_{sat}^T A^T)(r_{tag} - Ar_{sat}))^{\frac{1}{2}} \\
r &= (r_{tag}^T r_{tag} - r_{tag}^T Ar_{sat} - r_{sat}^T A^T r_{tag} + r_{sat}^T A^T Ar_{sat})^{\frac{1}{2}} \\
r &= (r_{tag}^T r_{tag} - 2r_{tag}^T Ar_{sat} + r_{sat}^T r_{sat})^{\frac{1}{2}} \\
\dot{r} &= \frac{1}{2||r_{tag} - Ar_{sat}||}(r_{tag}^T \dot{r}_{tag} + \dot{r}_{tag}^T r_{tag} - 2(\dot{r}_{tag}^T Ar_{sat} \\
&\quad + r_{tag}^T \dot{A} r_{sat} + r_{tag}^T A\dot{r}_{sat}) + r_{sat}^T \dot{r}_{sat} + \dot{r}_{sat}^T r_{sat}) \\
\dot{r} &= \frac{1}{2||r_{tag} - Ar_{sat}||}(r_{tag}^T \dot{r}_{tag} + \dot{r}_{tag}^T r_{tag} - 2\dot{r}_{tag}^T Ar_{sat} \\
&\quad - 2r_{tag}^T \dot{A} r_{sat} - 2r_{tag}^T A\dot{r}_{sat} + r_{sat}^T \dot{r}_{sat} + \dot{r}_{sat}^T r_{sat}) \\
\dot{r} &= \frac{1}{2||r_{tag} - Ar_{sat}||}(r_{tag}^T v_{tag} + v_{tag}^T r_{tag} - 2v_{tag}^T Ar_{sat} \\
&\quad - 2r_{tag}^T \dot{A} r_{sat} - 2r_{tag}^T Av_{sat} + r_{sat}^T v_{sat} + v_{sat}^T r_{sat})
\end{aligned} \tag{A.5}$$

Here the term $\dot{A}$ is the derivative of a rotation matrix which is equivalent to $A(\hat{\omega})$, where ω is the angular velocity of the Earth and the hat operator is the skew-symmetric matrix operator. Next, the derivation for time derivative of the ionospheric delay ($\dot{I}$)

is shown in (A.6).

$$
\begin{aligned}
I(\zeta) &= \frac{40.3TECV}{f^2}[1-(\frac{R_E\sin\zeta}{R_E+h_I})^2]^{\frac{-1}{2}} \\
I(\zeta) &= \frac{40.3TECV}{f^2}(\frac{-1}{2})([1-(\frac{R_E\sin\zeta}{R_E+h_I})^2]^{\frac{-3}{2}}(-2(\frac{R_E\sin\zeta}{R_E+H_I}))(\frac{(R_E+h_I)(R_E(\cos\zeta)\dot{\zeta})}{(R_E+h_I)^2})) \\
I(\zeta) &= \frac{40.3TECV}{f^2}\frac{(\frac{R_E\sin\zeta}{R_E+h_I})(\frac{(R_E+h_I)(R_E(\cos\zeta)\dot{\zeta})}{(R_E+h_I)^2})}{[1-(\frac{R_E sin\zeta}{R_E+h_I})^2]^{\frac{3}{2}}} \\
\dot{I}(\zeta) &= \frac{(40.3TECV)(\frac{(R_E\sin\zeta)(R_E(\cos\zeta)\dot{\zeta})}{(R_E+h_I)^2})}{f^2(1-(\frac{R_E\sin\zeta}{R_E+h_I})^2)^{\frac{3}{2}}}
\end{aligned}
\tag{A.6}
$$

A.3 Random Variable Transformations

Two important properties of a random variable (X) are the following:

$$
\begin{aligned}
E[cX] &= cE[X] \\
Var[cX] &= c^2Var[X]
\end{aligned}
\tag{A.7}
$$

The random variable is represented by a Gaussian and we use these properties to transform this Gaussian distribution with mean μ_1 and variance σ_1^2 into a Gaussian with a desired mean and variance to capture the effects of the actual variable. Suppose the initial random variable is sampled by the following normal distribution

$$
X \sim N(\mu_1, \sigma_1^2) \tag{A.8}
$$

and we want to transform the random variable X by a multiplicative constant (a) and additive scalar (b) to obtain a desired mean μ_2 and standard deviation σ_2^2. The transformation is $Y = aX + b$ and the derivation for updated mean and variance are

shown in equation (A.9).

$$\begin{aligned} Y &= aX + b \\ E[Y] &= E[aX] + E[b] \\ E[Y] &= aE[X] + b \\ E[Y] &= a\mu_1 + b \\ Var(Y) &= Var(aX) + Var(b) \\ Var(Y) &= a^2 Var(X) \end{aligned} \tag{A.9}$$

The constants a and b are then computed by solving the system of equations in (A.10).

$$\begin{aligned} a\mu_1 + b &= \mu_2 \\ a^2\sigma_1^2 &= \sigma_2^2 \end{aligned} \tag{A.10}$$

A.4 Linear Covariance Propagation

Once the nonlinear equation is linearized about a reference state, the jacobian (A) of the linear equation ($\Delta y = A\Delta x$) is used to propagate the uncertainty of the inputs to the outputs in the form of a covariance matrix. The derivation of (3.10) is shown below.

$$\begin{aligned} \Sigma_{\Delta y} &= E((\Delta y - \Delta\hat{y})(\Delta y - \Delta\hat{y})^T) \\ \Sigma_{\Delta y} &= E((A\Delta x - A\Delta\hat{x})(A\Delta x - A\Delta\hat{x})^T) \\ \Sigma_{\Delta y} &= E((A\Delta x - A\Delta\hat{x})(\Delta x^T A^T - \Delta\hat{x}^T A^T)) \\ \Sigma_{\Delta y} &= E(A(\Delta x - \Delta\hat{x})(\Delta x^T - \Delta\hat{x}^T)A^T) \\ \Sigma_{\Delta y} &= A(E((\Delta x - \Delta\hat{x})(\Delta x - \Delta\hat{x})^T))A^T \\ \Sigma_{\Delta y} &= A\Sigma_{\Delta x}A^T \end{aligned} \tag{A.11}$$

A.5 Numerical Stability

The condition number is a metric used for numerical analysis that indicates whether a matrix is well conditioned or ill conditioned. It is a measure of stability or sensitivity

of a matrix to numerical operations. It is defined as the ratio between the largest singular value (σ_{max}) and the smallest singular value (σ_{min}) of the matrix.

$$cond(A) = \frac{\sigma_{max}}{\sigma_{min}} \tag{A.12}$$

The condition number also indicates loss of precision of a linear system ($Ax = b$) solve as the number of decimals lost in the solution (x) is proportional to the base 10 log of the condition number as shown in (A.13).

$$decimals\ lost = log_{10}(cond(A)) \tag{A.13}$$

A condition number close to 1 indicates a well conditioned matrix while values significantly larger characterize a ill-conditioned matrix. For this project, the system from the root-finding problem was poorly conditioned and custom units had to be implemented to allow the system to solve along the entire horizon. The distance was scaled by the radius of the Earth, and the time was scaled to ensure the speed of light (c) was equal to 1.

A.

Bibliography

[1] CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers. 3.7

[2] Argos User's Manual, June 2016. 1, 1

[3] K. Aksnes, P. H. Andersen, and E. Haugen. A precise multipass method for satellite Doppler positioning. *Celestial Mechanics*, 44(4):317–338, December 1988. ISSN 0008-8714, 1572-9478. doi: 10.1007/BF01234271. URL http://link.springer.com/10.1007/BF01234271. 3.5, 3.6

[4] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1):65–98, January 2017. ISSN 0036-1445, 1095-7200. doi: 10.1137/141000671. URL https://epubs.siam.org/doi/10.1137/141000671. 2.3.6

[5] Byron D. Tapley, Bob E. Shutz, and George H. Born. *Statistical Orbit Determination*. Elsevier Academic Press, 2004. 2.3.4

[6] J Russell Carpenter and Christopher N D'Souza. Navigation Filter Best Practices. 2.3.4

[7] Robert J Danchik. An Overview of Transit Development. *JOHNS HOPKINS APL TECHNICAL DIGEST*, 19(1), 1998. 1

[8] Simon Danisch and Julius Krumbiegel. Makie.jl: Flexible high-performance data visualization for Julia. *Journal of Open Source Software*, 6(65):3349, 2021. doi: 10.21105/joss.03349. URL https://doi.org/10.21105/joss.03349. 3.8

[9] Chuong B Do. The Multivariate Gaussian Distribution. 2.2.1

[10] Ac Dragon, A Bar-Hen, P Monestiez, and C Guinet. Comparative analysis of methods for inferring successful foraging areas from Argos and GPS tracking data. *Marine Ecology Progress Series*, 452:253–267, April 2012. ISSN 0171-8630, 1616-1599. doi: 10.3354/meps09618. URL http://www.int-res.com/abstracts/meps/v452/p253-267/. 3.1

[11] Duncan Eddy. SatelliteDynamics.jl. URL https://github.com/sisl/SatelliteDynamics.jl/blob/master/docs/src/index.md. 2.3.6

[12] Giusy Falcone, Jacob B. Willis, and Zachary Manchester. Propulsion-Free Cross-Track Control of a LEO Small-Satellite Constellation with Differential Drag, June 2023. URL http://arxiv.org/abs/2306.13844. arXiv:2306.13844 [cs, eess]. 3.7

[13] Shashi Gowda, Yingbo Ma, Alessandro Cheli, Maja Gwóźzdź, Viral B. Shah, Alan Edelman, and Christopher Rackauckas. High-performance symbolic-numerics via multiple dispatch. *ACM Commun. Comput. Algebra*, 55(3):92–96, jan 2022. ISSN 1932-2240. doi: 10.1145/3511528.3511535. URL https://doi.org/10.1145/3511528.3511535. A.1

[14] Elizabeth F. Hartel, Wendy Noke Durden, and Greg O'Corry-Crowe. Testing satellite telemetry within narrow ecosystems: nocturnal movements and habitat use of bottlenose dolphins within a convoluted estuarine system. *Animal Biotelemetry*, 8(1):13, December 2020. ISSN 2050-3385. doi: 10.1186/s40317-020-00200-4. URL https://animalbiotelemetry.biomedcentral.com/articles/10.1186/s40317-020-00200-4. 1

[15] John T. Betts. *Practical Methods for Optimal Control and Estimation using Nonlinear Programming*. Society for Industrial and Applied Mathematics (SIAM), second edition edition, 2010. 3.2.2

[16] P. Kaminski, A. Bryson, and S. Schmidt. Discrete square root filtering: A survey of current techniques. *IEEE Transactions on Automatic Control*, 16(6):727–736, December 1971. ISSN 0018-9286. doi: 10.1109/TAC.1971.1099816. URL http://ieeexplore.ieee.org/document/1099816/. 2.3.5

[17] Kenneth Alan Myers. *Filtering Theory Methods and Applications to the Orbit Determination Problem for near-Earth Satellites*. PhD thesis, The University of Texas at Austin, 1974. 2.3.6

[18] Robert MacCurdy, Rich Gabrielson, Eric Spaulding, Alejandro Purgue, Kathryn Cortopassi, and Kurt Fristrup. Automatic Animal Tracking Using Matched Filters and Time Difference of Arrival. *Journal of Communications*, 4(7):487–495, August 2009. ISSN 1796-2021. doi: 10.4304/jcm.4.7.487-495. URL http://ojs.academypublisher.com/index.php/jcm/article/view/24. 3.1

[19] Robert B MacCurdy, Richard M Gabrielson, and Kathryn A Cortopassi. AUTOMATED WILDLIFE RADIO TRACKING. 1, 3.6

[20] Oliver Montenbruck and Eberhard Gill. Ionospheric Correction for GPS Tracking of LEO Satellites. *Journal of Navigation*, 55(2):293–304, May 2002. ISSN 0373-4633, 1469-7785. doi: 10.1017/S0373463302001789. URL https://www.cambridge.org/core/product/identifier/S0373463302001789/type/journal_article. 3.5

[21] P. Mukhtarov, D. Pancheva, B. Andonov, and L. Pashova. Global TEC maps

based on GNSS data: 1. Empirical background TEC model: 1. EMPIRICAL BACKGROUND TEC MODEL. *Journal of Geophysical Research: Space Physics*, 118(7):4594–4608, July 2013. ISSN 21699380. doi: 10.1002/jgra.50413. URL http://doi.wiley.com/10.1002/jgra.50413. 3.5

[22] Kenneth A. Myers and Byron D. Tapley. Dynamical Model Compensation for Near-Earth Satellite Orbit Determination. *AIAA Journal*, 13(3):343–349, March 1975. ISSN 0001-1452, 1533-385X. doi: 10.2514/3.49702. URL https://arc.aiaa.org/doi/10.2514/3.49702. 2.3.4, 2.3.4

[23] National Ocean and Atmospheric Administration and National Aeronautics and Space Administration. COSPAS-SARSAT Search and Rescue System. 1

[24] Oliver Montenbruck and Eberhard Gill. *Satellite Orbits*. Springer, 2012. 2.2.2, 2.2.3, 2.2.3, A.1

[25] Pratap Misra and Per Enge. *Global Positioning System: Signals, Measurements and Performance*. Ganga-Jamuna Press, second edition edition, 2006. 1, 2.3.2, 3.2.1, 3.5, 3.6

[26] Samik Raychaudhuri. Introduction to Monte Carlo simulation. In *2008 Winter Simulation Conference*, pages 91–100, Miami, FL, USA, December 2008. IEEE. ISBN 978-1-4244-2707-9. doi: 10.1109/WSC.2008.4736059. URL https://ieeexplore.ieee.org/document/4736059/. 3.3.1

[27] C. Rush. Ionospheric radio propagation models and predictions–A mini-review. *IEEE Transactions on Antennas and Propagation*, 34(9):1163–1170, September 1986. ISSN 0096-1973. doi: 10.1109/TAP.1986.1143951. URL http://ieeexplore.ieee.org/document/1143951/. 3.5

[28] B. D. Tapley and B. E. Schutz. Estimation of unmodeled forces on a lunar satellite. *Celestial Mechanics*, 12(4):409–424, December 1975. ISSN 0008-8714, 1572-9478. doi: 10.1007/BF01595388. URL http://link.springer.com/10.1007/BF01595388. 2.3.4

[29] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. Intelligent robotics and autonomous agents. MIT Press, Cambridge, Mass, 2005. ISBN 978-0-262-20162-9. OCLC: ocm58451645. 2.3.2

[30] Kevin Tracy. A Square-Root Kalman Filter Using Only QR Decompositions, August 2022. URL http://arxiv.org/abs/2208.06452. arXiv:2208.06452 [cs, eess]. 2.3.5

[31] James Richard Wertz and Wiley J. Larson, editors. *Space mission analysis and design*. Space technology library. Microcosm ; Kluwer, El Segundo, Calif. : Dordrecht ; Boston, 3rd ed edition, 1999. ISBN 978-0-7923-5901-2 978-1-881883-10-4. 2.3.4

[32] M.J. Witt, S. Åkesson, A.C. Broderick, M.S. Coyne, J. Ellick, A. Formia, G.C. Hays, P. Luschi, Stedson Stroud, and B.J. Godley. Assessing accuracy and utility of satellite-tracking data using Argos-linked Fastloc-GPS. *Animal Behaviour*, 80(3):571–581, September 2010. ISSN 00033472. doi: 10.1016/j.anbehav.2010.05.022. URL https://linkinghub.elsevier.com/retrieve/pii/S0003347210002113. 1

[33] James R Wright. Optimal Orbit Determination. 2.3.3

[34] Y.G. Zurabov, K.K. Ivanov, and A.D. Kuropyatnikov. COSPAS-SARSAT satellite system. In *3rd International Conference on Satellite Communications (IEEE Cat. No.98TH8392)*, pages 156–158, Moscow, Russia, 1998. Int. Centre for Sci. & Tech. Inf. ISBN 978-5-93184-002-4. doi: 10.1109/ICSC.1998.741402. URL http://ieeexplore.ieee.org/document/741402/. 1