The Pennsylvania State University
The J. Jeffrey and Ann Marie Fox Graduate School

A RECEDING HORIZON BASED SENSOR TASKING FOR TRACKING SPACE OBJECTS IN CISLUNAR REGIME

A Thesis in
Aerospace Engineering
by
Vishnu Tej Dasari

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2024

The thesis of Vishnu Tej Dasari was reviewed and approved by the following:

Puneet Singla
Harry and Arlene Schell Professor of Engineering
Department of Aerospace Engineering
Thesis Advisor

Roshan T. Eapen
Assistant Professor of Aerospace Engineering

Robert G. Melton
Professor of Aerospace Engineering

Amy R. Pritchett
Professor of Aerospace Engineering
Department Head

Abstract

Cislunar space is a growing area of interest as several countries plan missions to explore the moon and its surrounding regions for scientific, economical and national security reasons. With this increased interest, there is an apt need to accurately track spacecraft in the cislunar regime. This thesis develops an optimal sensor tasking approach to track space objects in the cislunar regime with the help of space-based sensors. The developed approach utilizes the non-product quadrature method known as the Conjugate Unscented Transform (CUT) to accurately and efficiently propagate the state uncertainty through the chaotic Circular Restricted Three Body Problem (CR3BP). The CUT approach further leads to the design of a higher order Kalman filter to fuse the model predictions with sensor observations.

An information theoretic metric, Mutual Information (MI), also known as the Kullback-Leibler divergence metric, is used to assess the quality of a sensor observation. The MI provides the ratio between the prior and posterior probability density functions, which gives a quantitative assessment of the reduction in uncertainty due to sensing decisions. An exhaustive search algorithm is used to maximize mutual information over all objects, sensors, and time steps for a given problem. Given the combinatorial growth associated with the exhaustive search, sub-optimal sequential algorithms are designed while exploiting the submodularity property of the MI metric. Different variants of the sequential algorithms (e.g. receding horizon window or sequential-in-time) are discussed depending on the size of the problem. A receding horizon approach solves the sensor tasking problem for a given subset of time steps before moving on to the next subset. This is done until the entire time span has been solved, meaning the sensors are tasked for each time step of the time span.

Numerical results are shown for the receding horizon window applied to two examples of an Earth-based space object tracking problem and a cislunar object tracking problem. The results show that the implementation of the receding horizon window approach was a success as the tasking provides the desired reduction in uncertainty of all objects over the time span. Specifically, the state errors are statistically analyzed and shown to not significantly impact the object tracking over time.

Table of Contents

List of Figures

List of Tables

Acknowledgments

This thesis has been a culmination of two years of hardwork and struggle. Although facing many ups and downs, I am proud of what I have accomplished so far.

I would like to thank my thesis advisor, Dr. Puneet Singla for introducing me to state estimation and information theory. He has been a wonderful teacher and advisor, being understanding and helpful in many situations, whether that be questions about my research or questions about my career after the degree. His courses on system analysis and system identification were also very informative and helpful, not necessarily always directly applicable to my research but general knowledge about other aspects of the field.

I would also like to thank the other two members on my committee being Dr. Roshan T. Eapen and Dr. Robert G. Melton. Dr. Eapen has been very helpful in coursework and discussing my research when I had questions. His courses on advanced orbital mechanics and modern computational astrodynamics were insightful and challenging and made me love astrodynamics even more. He also was very helpful when I had questions about the cislunar example he had done previous work on and was very open to listening to how I was solving the problem currently and working through the methodology with me. As for Dr. Melton, he was my introduction to astronautics and astrodynamics at Penn State. His courses were the reason I fell in love with the subjects in the first place, not just because of subject matter but also because he was a great professor. I would like to thank Pennsylvania State University for providing me a great education and the right tools to be successful regardless of my career choices in the future.

I would like to thank my family for all the support they have been providing me through this degree. First, I would like to thank my mother, Laxmikala Dasari, for her constant belief that I can achieve anything I set my mind to and always making me delicious home-cooked food when I went home. Next, I would like to thank my father, Venkata Ramesh Dasari, for both emotional and financial support throughout my degree. Regardless of any problems he might have been facing, he never let that affect me and would always get me anything I needed. Finally, I would like to thank my brother, Surya Tej Dasari, for taking care of our parents while I was pursuing a Master's degree. Even if our family was in financial troubles, he took it upon himself to take care of everything, even working two jobs and 60+ hours per week, but did not let me worry about anything besides my Master's degree.

I would like to thank my friends for providing me with constant support and making unforgettable memories. I would like to thank Goody for being an amazing roommate and

always believing in me. He literally survived a car crash but never stopped supporting me. I would like to thank Jeremy for always talking with me through any problems I had and playing World of Tanks with me when I needed a break from research. I would like to thank Julia for believing I could do anything and for being supportive regardless of how annoying I could be. I would like to thank Wyatt, Parker, Konner, Dartagnan, Zeke, and Jake for providing me with some great memories and helping me relax whenever I would get stressed. I would like to thank Anthony, Soph, Monsour, Shiv, Katie, and Killian for "distracting" me from research by constantly asking me questions about their homework. Finally, I would like to thank Isaac, Kokkis, John, Eric, Andy, Chris, and Gage for providing me with much-needed breaks by playing a kids' trading card game with me and traveling to various places to get shiny cardboard.

This material is based upon work supported jointly by the AFOSR grants FA9550-22-1-0092 and FA9550-23-1-0512. The findings and conclusions of this work do not necessarily reflect the views of the funding agencies.

Chapter 1
Introduction

1.1 Introduction

Since the Apollo era, space has become an increasingly valuable domain for national security, diplomatic, informational, and economic reasons. Space research has spurred socio-economic development, enabled scientific discoveries and advanced communications, remote sensing, geophysical, and astrophysical applications. The number of space objects launched within the last few years has grown exponentially. Missions that yield successful lunar resource extraction, including water, helium-3, regolith, and rare earth metals, will be foundational to the future multi-trillion-dollar space economy [2]. Establishing infrastructure in cislunar space also allows for increased surveillance capabilities and utilizing a currently unmonitored zone for accomplishing national security objectives. It is clear that space is emerging as an important civil, commercial, and military domain, and nations and industry partners are taking notice. The "New Space Race" has begun.

While only two nations had hopes of operating in the cislunar realm during the Apollo era, several nations and private industries are aiming for the Moon in the Artemis era. China launched the 2nd most satellites in 2021, outside of Starlink, and is planning to launch a constellation of satellites to support lunar exploration [3]. Counterspace capabilities allow militaries to either defend their space assets or attack others' assets, and the number of these counterspace tests and incidents increased dramatically within the past five years [4]. Similarly, the Artemis program has been envisioned with the intention of exploring cislunar space with long-term interest in conducting scientific analysis on the lunar landscape. Currently, plans for up to 10 missions have been proposed to the moon within the next decade [5]. In near-Earth regimes, the Orbital Sustainability Act of 2023 [6] directs specified agencies to take actions to remediate orbital debris in Earth orbit. However, as the number of missions increase, a natural increase in the number of

spacecraft and debris in cislunar space will also occur. If this growth is left unchecked, then the number of these objects will become unmanageable. Hence, there is a need to extend space situational awareness (SSA) capabilities beyond the geosynchronous Earth orbit (GEO) domain, i.e., XGEO.

The perturbed two-body (Keplerian) framework has led to extensive modeling, providing various approximate solutions of increasing fidelity and analysis of representative behaviors to investigate spacecraft motion in orbits around Earth. The propagation of state errors through the orbit dynamical models and navigation in orbits around Earth are well studied [1, 7–11]. However, beyond GEO (XGEO), the fundamental structure of space trajectories can be and typically are radically different. Potential orbits and trajectories do not simply consist of the conic sections that typiy orbits nearer to Earth. The challenges that limit the transferability of tools and techniques from the GEO to XGEO region are primarily due to non-Keplerian dynamics, data sparsity from limited coverage, and frequent unavailability of sensors for absolute and relative navigation. While conic sections in the two-body problem enable relatively straightforward initial orbit determination (IOD) for a variety of measurement types in GEO, there is currently no known IOD analog for the cislunar regime, even for the foundational circular restricted three body problem (CR3BP). Currently, for near-Earth orbits, spacecraft orbits are characterized using traditional Two-Line Elements (TLEs). However, this TLE-based structure is not adequate to characterize cislunar orbits. It is entirely possible that a global structure equivalent to TLEs does not even exist for the CR3BP framework. In addition, outside the range of the Global Positioning System (GPS), it is a challenging affair to get accurate guidance and navigation capabilities in cislunar space. Gravitational acceleration in cislunar space from the Earth and moon is typically an order of magnitude smaller than in low Earth orbit (LEO). For certain situations, this is beneficial since less propulsion is needed to make large state changes. For example, the Δv, or change in velocity needed to make a trajectory maneuver, required to move from the lunar surface to a low lunar orbit (LLO) is approximately one-fifth of the Δv required to move from the Earth's surface to a LEO [12]. However, this simple fact also means an increased sensitivity to state errors once on a designated nominal trajectory. The presence of errors in models, navigation, and control can more easily cause significant disturbances.

Monte Carlo (MC) simulations can be used to propagate the uncertainty in the state vector through the nonlinear system [13]. However, these simulations are computationally expensive since the convergence of the MC is inversely proportional to the square root of the number of sample points. This means that, to gain one decimal place accuracy,

the number of samples needs to be increased a hundredfold. This leads to pursuing deterministic methods, such as the Gaussian quadrature scheme. However, this method suffers from the *curse of dimensionality* as the number of samples required grows exponentially. The Conjugate Unscented Transform (CUT) of the Kalman Filter is a non-Gaussian quadrature scheme that has been previously successfully used for various SSA applications, including cislunar applications [1, 14].

To understand the necessity of the CUT filter, the general state estimation methodology as applied to a cislunar application must be understood. Starting from the dynamical model, "accurately" representing the system physics is not a simple task as discussed with the high sensitivities to errors in the state. To help this cause, different sensors can be used based on the application to predict the current state of the physical system. However, simply relying on sensor data can be inaccurate as noise can corrupt the quality of the measurements, once again introducing uncertainty. This means that regardless of the approach, uncertainty must be considered in the model simulation. Typically, uncertainty is modeled using probability density functions (PDF), which allows for propagation of the uncertainty to be governed by the Chapman-Kolmogorov Equation (CKE) for a discrete-time system and the Fokker Planck Kolmogorov Equation for a continuous system [15]. For this thesis, discrete time systems will be considered and the PDFs will be Gaussian. Using Bayes' theorem [15] along with the CKE, the propagation of the uncertainty can be accurately modeled and sensor data can be fused with the expected state model to generate an accurate update of the state PDF.

For a given linear state model, the optimal solution is given by the Kalman Filter [16]. However, cislunar space cannot be approximated well as a linear system similar to many other real world applications, hence a solution methodology for nonlinear systems must be developed. Since obtaining exact solutions for nonlinear systems is usually practically infeasible or impossible, approximate solutions have been developed. Several different solution frameworks have been developed, including the Extended Kalman Filter (EKF) [17], which involves using the Jacobian to linearize the nonlinear state model and finding the optimal solution to the linearized problem. However, this approach only provides an accurate answer for systems that the linearization provides a good approximation for, so different methods must be explored for the cislunar case. Using the Gaussian PDF to represent the uncertainty, the problem can be transformed into evaluating expectation integrals that appear in the definitions of the statistical moments of the PDF. The primary method of computing these expectation integrals is the Unscented Transform (UT) of the Kalman Filter [18]. This method involves generating

points that reproduce the statistical moments of the PDF up to 3rd order. This means that highly nonlinear systems, such as the cislunar model, will yield large errors as the approximation does not accurately capture the nonlinearity. Hence, the UT will also not work for cislunar state estimation purposes. To tackle these highly nonlinear systems, an extension of the UT was developed known as the CUT. This extends the UT to higher orders, the highest being CUT8, which is accurate up to 8th order statistical moments.

Assuming the data fusion with the state model can accurately predict the state vector, the next challenge that needs to be addressed is sensor tasking and measurement quality. For a given sensor, the sensor parameters can be modified such that measurements can be made. If considering the problem of sensors being used to track space targets, modeling of sensors typically involves defining a field of view (FOV) that limits the area in which a measurement can be made on a given target. Additional constraints in the form of distance from the sensor to the target can be enforced to affect the detection probability of the target. This is applicable to cislunar space as there are significantly larger distances when comparing to the GEO domain, meaning the detection probability of a target can be severely affected. Given these constraints, pointing a sensor in only one direction could lead to high uncertainty in a target's state vector as determined by the nature of the target's orbit. Certain orbits could cause a target to be seen for a short time and not be seen for a significant portion of time after the initial measurement, which will lead to large uncertainties. This problem leads to the natural solution of dynamically reorienting sensors to keep targets within the field of view for longer, thus reducing the overall uncertainty for the target's state vector. However, constantly reorienting one sensor to track one target is not an efficient allocation of resources since the number of possible targets far exceed the number of sensors available for tracking. So with the goal of tracking multiple targets using a select few sensors, a sensor performance metric can be introduced to determine an optimal method for tasking sensors to track certain targets at certain times.

The Fisher Information Matrix (FIM) is one sensor performance metric that can be utilized to provide the information gained by a sensor tracking a target. The FIM is the inverse of the Cramer-Rao Lower Bound (CRLB), which is the lower bound of the covariance matrix. This means that maximizing the FIM is desired. However, the FIM is obtained for nonlinear systems through linearizations, meaning the information is obtained using an estimate of the uncertainty [19]. So, for a cislunar case, the FIM might not properly define the information in the system. Thus, the concept of mutual information can be used to develop a sensor scheduling algorithm [20]. The general idea

of the mutual information (MI) approach involves comparing the prior and posterior covariance for a given sensor tasking schedule and maximizing the reduction between the two covariance matrices, or minimizing the posterior covariance. The sensor schedule that minimizes the posterior covariance is considered to be the optimal solution. This MI approach can be applied to tracking the targets in cislunar space given a select few sensors for a given discretized time span, as the CUT Filter allows for the nonlinearities in cislunar space to be taken into account accurately. Typically, the MI approach solves the problem at each step of the time span. However, a receding horizon window approach can be implemented, which creates subsets of time steps in the time span and solves the problem for each of those subsets before moving on to the next one.

1.2 Objective of Thesis

The main objective of this thesis is to utilize CUT based uncertainty propagation in conjunction with information theory to task space based sensors to accurately track targets in the cislunar regime. In this respect, the previous work [14] will be leveraged and adopted for cislunar target tracking.

The thesis first discusses the theory behind the sensor tasking algorithm and then apply it to specific situations. Chapter 2 formulates the Kalman Filter, both the linear and nonlinear versions, starting from first principles. As it is typically infeasible or impossible to obtain the exact solutions for a nonlinear system, approximate solution frameworks are introduced. The concept of sigma points is discussed and the merits of using the CUT version of the Kalman Filter compared to the UT version are shown with an example. Specifically, the example compares the UT, which is 3rd order accurate, to CUT4, CUT6, and CUT8, which are 4th, 6th, and 8th order accurate, respectively.

Chapter 3 begins by defining mutual information generally as a sensor performance metric and explaining how the concept of mutual information relates to sensor tasking. Then, the expression for mutual information is formulated and the computation of the joint mutual information for a system is discussed. Sub-optimal methods are discussed to alleviate computational expense for minor accuracy reduction. Examples are shown for both an Earth and cislunar space target tracking example and a general discussion of the results is be provided.

Chapter 4 summarizes the contents of this thesis and provide general ideas for future work.

Chapter 2
Dynamical System State Estimation

2.1 Introduction to State Estimation

The primary objective of state estimation is to estimate the state of a dynamical system through the fusion of dynamical model prediction with sensor observation while accounting for errors in both. Dynamical models are often an approximate representation of a physical process and hence have uncertainties associated with them known as process noise. Generally, they are derived making use of physical laws, such as Newton's laws of motion, and the model error can result from several sources, such as approximating the underlying physics or unknown disturbances or parameters. The use of sensor data to correct and refine the dynamical model prediction to reduce the associated uncertainties is a logical improvement over purely model-based prediction. However, mathematical models for various sensors are generally based upon the "usefulness" rather than the "truth." Hence, not all information that one would like to know is provided.

To accurately capture the system state uncertainty, probabilistic means are explored. More specifically, a PDF is assumed to quantify the uncertainty in the state model, along with the process noise and measurement noise. Given the prior information about the initial state of the system, the dynamical model is used to propagate the state uncertainty until the observation time and then a Bayesian formulation is used to fuse model prediction with the observation data. The Bayesian approach had its birth with the development of the Kalman Filter [16].

Using the state, measurement, and noise info, an estimate of the state vector at each time step can be determined by generating a posteriori PDF. The a posteriori PDF at time step k can be considered as the a priori PDF at time step $k+1$, and the process can

be iterated through the entire time span. This process lends itself to a filtering algorithm.

In the following sections of Chapter 2, the derivation of the Kalman filter update equations for both linear and nonlinear systems will be shown. However, analytically computing the nonlinear update equations is often unviable and necessitates developing an approximate solution. Thus, a general outline for applying the Kalman filter at each time step will be shown along with discussing various approximate solution frameworks and the general similarities/differences. The chapter will conclude with providing an approximate solution framework known as the CUT formulation for the Kalman filter, as well as implementing the filter to an example problem to showcase the benefits of the higher-order filter.

2.2 Maximum Likelihood Estimate

Before deriving the Kalman update equations, one can derive an optimal solution for the state vector given solely sensor measurement data. This can be achieved by maximizing the likelihood function, $p(\mathbf{y}|\mathbf{x})$, which describes the probability of observation data according to a given measurement model

$$\max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}) \equiv \max_{\mathbf{x}} \ln p(\mathbf{y}|\mathbf{x}) \tag{2.1}$$

where $\mathbf{y} \in \mathbb{R}^m$ is the measurement vector, $\mathbf{x} \in \mathbb{R}^n$ is the state vector, and $p(\mathbf{y}|\mathbf{x})$ is the conditional PDF that represents the likelihood of $\mathbf{y}$ given $\mathbf{x}$. The measurement model for $\mathbf{y}$ is assumed to be linear, i.e.,

$$\mathbf{y} = H\mathbf{x} + \boldsymbol{\nu} \tag{2.2}$$

where H is the linear measurement model, and $\boldsymbol{\nu}$ is the measurement noise modeled as a Gaussian random variable with zero mean and a covariance matrix R. This results in the following expression for $p(\mathbf{y}|\mathbf{x})$:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{\sqrt{2\pi|R|}} e^{-\frac{1}{2}(\mathbf{y}-H\hat{\mathbf{x}})^T R^{-1}(\mathbf{y}-H\hat{\mathbf{x}})} \tag{2.3}$$

A primary assumption is that no previous information is known so as to not introduce bias, meaning the PDF of the state will be assumed to be unknown. Now, substituting

for the likelihood function in (2.1) results in a weighted cost function J

$$\min_{\mathbf{x}} J = \frac{1}{2}(\mathbf{y} - H\hat{\mathbf{x}})^T R^{-1}(\mathbf{y} - H\hat{\mathbf{x}}) \tag{2.4}$$

where $\hat{\mathbf{x}}$ is the expected value of the state vector. The cost function J is subject to the necessary condition

$$\frac{\partial J}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}} = 0 \tag{2.5}$$

Using matrix properties and applying the derivative constraint, the cost function constraint in Equation (2.5) turns into

$$H^T R^{-1}\mathbf{y} = H^T R^{-1} H\mathbf{x} \tag{2.6}$$

Since $H^T R^{-1} H$ is a linearly independent square matrix, the inverse can be taken of the entire quantity to solve for the optimal $\mathbf{x}$

$$\mathbf{x} = (H^T R^{-1} H)^{-1} H^T R^{-1}\mathbf{y} \tag{2.7}$$

This result is defined to be the maximum likelihood estimate and is one method of computing the state vector at a given time step. The result provides the best minimum variance unbiased estimator for a linear system. However, if prior information exists about the state, one could achieve a better state estimate by including the information in the estimation process.

2.3 Maximum A-Posteriori Estimate

The maximum a-posteriori (MAP) estimate involves maximizing the posterior PDF

$$\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}) \equiv \max_{\mathbf{x}} \ln p(\mathbf{x}|\mathbf{y}) \tag{2.8}$$

According to Bayes' Theorem, the posterior PDF is proportional to the product of the likelihood and prior PDF. This effectively weights the likelihood using the prior PDF to produce a better state estimate using known information. The cost function shown in Equation (2.8) can be simplified using Bayes' Theorem.

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \tag{2.9}$$

where $p(\mathbf{x})$ is the prior PDF representing state uncertainty, $p(\mathbf{y})$ is the total probability of observation data of $\mathbf{y}$, and $p(\mathbf{x}|\mathbf{y})$ represents the likelihood of $\mathbf{x}$ given $\mathbf{y}$ or the posterior PDF. The denominator of Equation (2.9) can be written in integral form and substituted.

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{\int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x}} \tag{2.10}$$

Assuming the PDFs are Gaussian, expressions for the Gaussian PDF can be defined using the following $p(\mathbf{x}) \sim \mathcal{N}(\mathbf{x} : \hat{\mathbf{x}}, P)$ and $p(\mathbf{y}|\mathbf{x}) \sim \mathcal{N}(\mathbf{y} : H\hat{\mathbf{x}}, R)$. The full expressions for the two Gaussian PDFs are shown.

$$p(\mathbf{x}) = \frac{1}{\sqrt{2\pi|P|}} e^{-\frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})^T P^{-1}(\mathbf{x}-\hat{\mathbf{x}})} \tag{2.11}$$

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{\sqrt{2\pi|R|}} e^{-\frac{1}{2}(\mathbf{y}-H\mathbf{x})^T R^{-1}(\mathbf{y}-H\mathbf{x})} \tag{2.12}$$

Multiplying the two PDFs together produces

$$p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) = \frac{1}{\sqrt{2\pi|R|}} \frac{1}{\sqrt{2\pi|P|}} e^{-\frac{1}{2}(\mathbf{y}-H\mathbf{x})^T R^{-1}(\mathbf{y}-H\mathbf{x}) - \frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})^T P^{-1}(\mathbf{x}-\hat{\mathbf{x}})} \tag{2.13}$$

Consider the exponent in Equation (2.13).

$$D = -\frac{1}{2}(\mathbf{y} - H\mathbf{x})^T R^{-1}(\mathbf{y} - H\mathbf{x}) - \frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}})^T P^{-1}(\mathbf{x} - \hat{\mathbf{x}}) \tag{2.14}$$

D can be further simplified to the following expression.

$$D = -\frac{1}{2}(\mathbf{y}^T R^{-1}\mathbf{y} + \hat{\mathbf{x}}^T P^{-1}\hat{\mathbf{x}} + \mathbf{x}^T(H^T R^{-1}H + P^{-1})\mathbf{x} - 2(\mathbf{y}^T R^{-1}H + \hat{\mathbf{x}}^T P^{-1})\mathbf{x}) \tag{2.15}$$

From Equation (2.15), the following expressions can be defined

$$A = H^T R^{-1}H + P^{-1} \tag{2.16}$$

$$\mathbf{b}^T = \mathbf{y}^T R^{-1}H + \hat{\mathbf{x}}^T P^{-1} \tag{2.17}$$

Substituting Equations (2.16) and (2.17) into Equation (2.15), the exponent can be written as

$$D = -\frac{1}{2}(\mathbf{y}^T R^{-1}\mathbf{y} + \hat{\mathbf{x}}^T P^{-1}\hat{\mathbf{x}} + \mathbf{x}^T A\mathbf{x} - 2\mathbf{b}^T\mathbf{x}) \tag{2.18}$$

Going back to Bayes' theorem, by substituting Equation (2.18) into Equation (2.13) the numerator of the cost function can be written as

$$p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) = \frac{1}{\sqrt{2\pi|R|}}\frac{1}{\sqrt{2\pi|P|}}e^{-\frac{1}{2}(\mathbf{y}^T R^{-1}\mathbf{y}+\hat{\mathbf{x}}^T P^{-1}\hat{\mathbf{x}}+\mathbf{x}^T A\mathbf{x}-2\mathbf{b}^T\mathbf{x})} \tag{2.19}$$

Similarly, the denominator is the integral of the numerator and is given as

$$\int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int \frac{1}{\sqrt{2\pi|R|}}\frac{1}{\sqrt{2\pi|P|}}e^{-\frac{1}{2}(\mathbf{y}^T R^{-1}\mathbf{y}+\hat{\mathbf{x}}^T P^{-1}\hat{\mathbf{x}}+\mathbf{x}^T A\mathbf{x}-2\mathbf{b}^T\mathbf{x})}d\mathbf{x} \tag{2.20}$$

Taking out the terms that are constant with respect to $\mathbf{x}$ and evaluating the remaining integral, Equation (2.20) can be simplified to the following expression

$$\int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x} = \frac{\sqrt{2\pi|A^{-1}|}}{\sqrt{2\pi|R|}\sqrt{2\pi|P|}}e^{-\frac{1}{2}(\mathbf{y}^T R^{-1}\mathbf{y}+\hat{\mathbf{x}}^T P^{-1}\hat{\mathbf{x}}-\mathbf{b}^T A^{-T}\mathbf{b})} \tag{2.21}$$

Dividing the numerator in Equation (2.19) by the denominator in Equation (2.21), the posterior PDF becomes

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{\sqrt{2\pi|A^{-1}|}}e^{-\frac{1}{2}(\mathbf{x}-A^{-1}\mathbf{b})^T A(\mathbf{x}-A^{-1}\mathbf{b})} \tag{2.22}$$

Comparing the final expression in Equation (2.22) to a general Gaussian formula, it is clear that for the posterior PDF $p(\mathbf{x}|\mathbf{y})$, the mean vector and covariance are

$$\boldsymbol{\mu} = A^{-1}\mathbf{b} \tag{2.23}$$

$$\Sigma = A^{-1} \tag{2.24}$$

where μ is the posterior state mean and Σ is the posterior state covariance. Now, the matrix inversion lemma can be used to find the inverse of A. The matrix inversion lemma is as follows

$$G^{-1} = \left(B^{-1} + C^T DC\right)^{-1}$$
$$G^{-1} = B - BC^T(CBC^T + D^{-1})^{-1}CB$$

Applying this to A, the resultant expression with $B = P$, $C = H$, and $D = R^{-1}$ is

$$A^{-1} = P - PH^T(HPH^T + R)^{-1}HP \tag{2.25}$$

From Equation (2.25), the matrix K can be defined as

$$K = PH^T(HPH^T + R)^{-1} \tag{2.26}$$

Substituting into Equation (2.25)

$$A^{-1} = P - KHP \tag{2.27}$$

Equations (2.17) and (2.27) can be substituted into $\boldsymbol{\mu}$ in Equation (2.23)

$$\boldsymbol{\mu} = (P - KHP)(H^T R^{-1}\mathbf{y} + P^{-1}\hat{\mathbf{x}}) \tag{2.28}$$

Expanding Equation (2.28) and simplifying using matrix properties, the following expression is obtained.

$$\boldsymbol{\mu} = K\mathbf{y} + \hat{\mathbf{x}} - KH\hat{\mathbf{x}} \tag{2.29}$$

Now, the Kalman update equations for a linear system can be defined as

$$\boldsymbol{\mu} = \hat{\mathbf{x}} + K(\mathbf{y} - H\hat{\mathbf{x}}) \tag{2.30}$$

$$\Sigma = P - KHP \tag{2.31}$$

As mentioned previously, the update equations can be used with the measurement data at each time step k to update the prior PDF and obtain a posterior PDF. The posterior PDF can then be used as the prior PDF for the next time step $k+1$ and the process can be iterated through the entire time span.

2.4 Linear Uncertainty Propagation

Before proceeding to the nonlinear Kalman Update Equations, it is important to confirm that propagating a Gaussian PDF modeling uncertainty remains Gaussian, so that using the posterior PDF as the prior PDF in the next time step is valid.

Using the Chapman-Kolmogorov equation (CKE) shown, one can use the relation of

the prior and posterior state PDF to show that the posterior PDF does remain Gaussian.

$$p(\mathbf{x}_{k+1}) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k)d\mathbf{x}_k \tag{2.32}$$

For a discrete-time, linear system, the dynamical model can be represented by the following expression

$$\mathbf{x}_{k+1} = F_k\mathbf{x}_k + \boldsymbol{\omega}_k \tag{2.33}$$

where F_k is the constant, linear state matrix, $\mathbf{x}_k$ is generated from the Gaussian PDF $p(\mathbf{x}) \sim \mathcal{N}(\mathbf{x} : \hat{\mathbf{x}}, P)$, and $\boldsymbol{\omega}_k$ is generated from the Gaussian distribution $\boldsymbol{\omega}_k \sim \mathcal{N}(\boldsymbol{\omega}_k : \mathbf{0}, Q_k)$. The conditional likelihood function $p(\mathbf{x}_{k+1}|\mathbf{x}_k) = p(\mathbf{x}_{k+1} - F_k\mathbf{x}_k)$ for the linear system. Now, the full expression for the conditional likelihood function and the prior PDF can be written:

$$p(\mathbf{x}_k) = \frac{1}{\sqrt{2\pi|P_k|}} e^{-\frac{1}{2}(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T P_k^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_k)} \tag{2.34}$$

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k) = \frac{1}{\sqrt{2\pi|Q_k|}} e^{-\frac{1}{2}(\mathbf{x}_{k+1} - F_k\mathbf{x}_k)^T Q_k^{-1}(\mathbf{x}_{k+1} - F_k\mathbf{x}_k)} \tag{2.35}$$

Substituting the expressions into the CKE, the following expression is obtained.

$$p(\mathbf{x}_{k+1}) = \int \frac{1}{2\pi\sqrt{|P_k||Q_k|}} e^{-\frac{1}{2}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T P^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_k) + (\mathbf{x}_{k+1} - F_k\mathbf{x}_k)^T Q_k^{-1}(\mathbf{x}_{k+1} - F_k\mathbf{x}_k)\right]} d\mathbf{x}_k \tag{2.36}$$

The exponent, shown below,

$$D = -\frac{1}{2}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T P^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_k) + (\mathbf{x}_{k+1} - F_k\mathbf{x}_k)^T Q_k^{-1}(\mathbf{x}_{k+1} - F_k\mathbf{x}_k)\right] \tag{2.37}$$

can be expanded and grouped, similar to the previous section, such that new quantities can be defined, as shown

$$D = -\frac{1}{2}\left(\mathbf{x}_k^T B\mathbf{x}_k - 2\mathbf{g}^T\mathbf{x}_k + c\right) \tag{2.38}$$

where the terms B, $\mathbf{g}^T$, and c are defined in the following expressions.

$$B = P_k^{-1} + F_k^T Q_k^{-1} F_k \tag{2.39}$$

$$\mathbf{g}^T = \hat{\mathbf{x}}_k^T P_k^{-1} + \mathbf{x}_{k+1}^T Q_k^{-1} F_k \tag{2.40}$$

$$c = \mathbf{x}_{k+1}^T Q_k^{-1} \mathbf{x}_{k+1} + \hat{\mathbf{x}}_k^T P_k^{-1} \hat{\mathbf{x}}_k \tag{2.41}$$

Substituting in these definitions and taking out the constant terms and the c term from the integral since they do not depend on $\mathbf{x}_k$ explicitly, the following expression for $p(\mathbf{x}_{k+1})$ is obtained

$$p(\mathbf{x}_{k+1}) = \frac{1}{2\pi\sqrt{|P_k||Q_k|}} e^{\left(-\frac{1}{2}c\right)} \int e^{-\frac{1}{2}\left(\mathbf{x}_k^T B \mathbf{x}_k - 2\mathbf{g}^T \mathbf{x}_k\right)} d\mathbf{x}_k \tag{2.42}$$

The integral can now be taken and simplified into the following expression for $p(\mathbf{x}_{k+1})$

$$p(\mathbf{x}_{k+1}) = \frac{\sqrt{|B^{-1}|}}{\sqrt{2\pi|P_k||Q_k|}} e^{\left[-\frac{1}{2}\left(c + \mathbf{g}^T \left(B^T\right)^{-1} \mathbf{g}\right)\right]} \tag{2.43}$$

Finally, the definitions of B,$\mathbf{g}^T$, and c can be re-substituted and the expression for the propagated prior PDF can be simplified.

$$p(\mathbf{x}_{k+1}) = \frac{1}{\sqrt{2\pi|F_k P_k F_k^T + Q_k|}} e^{-\frac{1}{2}(\mathbf{x}_{k+1} - F_k \hat{\mathbf{x}}_k)^T \left(F_k P_k F_k^T + Q_k\right)^{-1} (\mathbf{x}_{k+1} - F_k \hat{\mathbf{x}}_k)} \tag{2.44}$$

Analyzing Equation (2.44), it is obvious that the propagated prior PDF remains Gaussian as it retains the general form of a Gaussian distribution, where the mean and covariance are shown below.

$$\hat{\mathbf{x}}_{k+1} = F_k \hat{\mathbf{x}}_k \tag{2.45}$$

$$P_{k+1} = F_k P_k F_k^T + Q_k \tag{2.46}$$

This confirms the validity of the step of making the posterior PDF at one time step equal to the prior PDF at the next time step.

2.5 Linear Kalman Filter Summary

The linear Kalman update equations were generated using the MAP formulation. These equations provide a basis for updating the state mean and covariance from one time step to the next given sensor measurement data for a linear state and measurement model. This process can be iterated through the entire time span for given sensor measurements. However, to confirm the propagated PDF can be used as the prior PDF of the next time step, the CKE was used to prove that the propagated PDF remains Gaussian and that the assumption is valid.

2.6 Minimum Variance Estimate

Another method of deriving the Kalman filter update equations is by minimizing the posterior variance. This differs from the previous method that maximized the posterior PDF. This is important as now, no assumptions about the measurement model have to be made, specifically the measurement model can be nonlinear. The result will provide the equivalent of the maximum likelihood estimate.

2.6.1 Defining the Nonlinear Problem

The minimization problem of the posterior variance can be setup as follows

$$\min_{\hat{\mathbf{x}}} Tr[E[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T]] \tag{2.47}$$

In Section 2.3, an expression for $\boldsymbol{\mu}$ was obtained assuming a linear measurement model. Removing that assumption would yield the following expression

$$\boldsymbol{\mu} = \hat{\mathbf{x}} + K(\mathbf{y}-\hat{\mathbf{y}}) \tag{2.48}$$

In Equation (2.48), the expression for K is unknown as the previous expression assumed a linear measurement model. Assuming $\mathbf{y} = h(\mathbf{x}) + \boldsymbol{\nu}$, a new general expression has to be derived. By substituting Equation (2.48) into the minimization problem in Equation (2.47), the following is obtained

$$\min_{K} Tr[E[((\mathbf{x}-\hat{\mathbf{x}}) - K(\mathbf{y}-\hat{\mathbf{y}}))((\mathbf{x}-\hat{\mathbf{x}}) - K(\mathbf{y}-\hat{\mathbf{y}}))^T]] \tag{2.49}$$

By applying properties of matrices and the expected value and by selectively grouping terms, the expression can be turned into

$$\min_{K} Tr[J] \tag{2.50}$$

where

$$\begin{aligned} J = {} & E[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^T] - KE[(\mathbf{y}-\hat{\mathbf{y}})(\mathbf{x}-\hat{\mathbf{x}})^T] - E[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{y}-\hat{\mathbf{y}})^T]K^T \\ & + KE[(\mathbf{y}-\hat{\mathbf{y}})(\mathbf{y}-\hat{\mathbf{y}})^T]K^T \end{aligned} \tag{2.51}$$

From this expression and the expected value definition of covariance, the following covariance matrices can be defined

$$P_x = E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T] \tag{2.52}$$

$$P_{xy} = E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{y} - \hat{\mathbf{y}})^T] \tag{2.53}$$

$$P_{yx} = E[(\mathbf{y} - \hat{\mathbf{y}})(\mathbf{x} - \hat{\mathbf{x}})^T] \tag{2.54}$$

$$P_y = E[(\mathbf{y} - \hat{\mathbf{y}})(\mathbf{y} - \hat{\mathbf{y}})^T] \tag{2.55}$$

These definitions can be substituted into Equation (2.50) turning it into

$$\min_K Tr[P_x - KP_{yx} - P_{xy}K^T + KP_yK^T] \tag{2.56}$$

Now the necessary condition can be implemented to Equation (2.56) to find an analytic expression for K

$$\frac{\partial J}{\partial K} = 0 \tag{2.57}$$

Substituting in Equation (2.56) to Equation (2.57), the following expression is obtained

$$\frac{\partial}{\partial K}[Tr[P_x - KP_{yx} - P_{xy}K^T + KP_yK^T]] = 0 \tag{2.58}$$

Using matrix properties and the facts that $P_{xy} = (P_{yx})^T$ and P_x does not depend on K, the derivative can be taken and the resultant equation is

$$-2P_{xy} + P_yK + K^TP_y = 0 \tag{2.59}$$

Furthermore, Equation (2.59) can be solved to obtain an analytical expression for K.

$$K = P_{xy}(P_y)^{-1} \tag{2.60}$$

As stated before, K does not assume a measurement model and thus is a generalized form of the expression. Now that a general expression for K has been found, the nonlinear covariance update equation for the Kalman Filter can be written generally as

$$\Sigma = P - KP_{xy}^T \tag{2.61}$$

Equations (2.48) and (2.61) together provide the framework for the updating process for the general, nonlinear Kalman Filter.

2.6.2 Linearization of the Problem

One method of simplifying the nonlinear solution of the Kalman Filter would be to linearize about the mean state vector. Although the measurement model is not considered to be linear, the general measurement equation can be linearized to

$$h(\mathbf{x}) = h(\hat{\mathbf{x}}) + H\Big|_{\mathbf{x}=\hat{\mathbf{x}}} (\mathbf{x} - \hat{\mathbf{x}}) \tag{2.62}$$

where the matrix H consists of the partials of the measurement model.

$$H = \frac{\partial h}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}} \tag{2.63}$$

Substituting Equation (2.62) into the expected value definitions for the cross-covariance and measurement covariance as shown in Equations (2.55) and (2.53), the following linearized definitions are obtained.

$$P_y = HPH^T + R \tag{2.64}$$

$$P_{xy} = PH^T \tag{2.65}$$

Substituting Equations (2.64) and (2.65) into Equation (2.60), the matrix K formulated for the linear Kalman Update Equations is retained, as shown in Equation (2.26). The update equations would use the nonlinear mean update from Equation (2.48) and the linear covariance update from Equation (2.31). The update equations formulated using the matrix H defined in Equation (2.63) are known as the EKF update equations.

2.6.3 Nonlinear Uncertainty Propagation

Analyzing the propagated state mean and covariance equations from Section 2.4, it is clear that the propagated mean and covariance are equivalent to the expectation value definitions of the posterior mean and covariance.

$$\hat{\mathbf{x}}_{k+1} = E\left[\mathbf{x}_{k+1}\right] = F_k \hat{\mathbf{x}}_k \tag{2.66}$$

$$P_{k+1} = E\left[\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}\right)\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}\right)^T\right] = F_k P_k F_k^T + Q_k \tag{2.67}$$

where $\mathbf{x}_{k+1} = F_k\mathbf{x}_k + \boldsymbol{\omega}_k$. For the nonlinear Kalman filter, the same propagation method can be followed and the assumption that the posterior PDF is Gaussian remains valid, however, the expectation values are general and cannot be simplified since the state model is nonlinear. The prior mean and covariance equations in expectation value form are:

$$\hat{\mathbf{x}}_{k+1|k} = E\left[f(\mathbf{x}_k)\right] \tag{2.68}$$

$$P_{k+1|k} = E\left[\left(f(\mathbf{x}_k) + \boldsymbol{\omega}_k - \hat{\mathbf{x}}_{k+1|k}\right)\left(f(\mathbf{x}_k) + \boldsymbol{\omega}_k - \hat{\mathbf{x}}_{k+1|k}\right)^T\right] \tag{2.69}$$

Equation (2.69) can be expanded and the expected value can be appropriately distributed to simplify the equation into

$$P_{k+1|k} = E\left[\left(f(\mathbf{x}_k) - \hat{\mathbf{x}}_{k+1|k}\right)\left(f(\mathbf{x}_k) - \hat{\mathbf{x}}_{k+1|k}\right)^T\right] + E\left[\boldsymbol{\omega}_k\boldsymbol{\omega}_k^T\right] \tag{2.70}$$

The last simplification that can be done is that the process noise covariance matrix can be substituted.

$$P_{k+1|k} = E\left[\left(f(\mathbf{x}_k) - \hat{\mathbf{x}}_{k+1|k}\right)\left(f(\mathbf{x}_k) - \hat{\mathbf{x}}_{k+1|k}\right)^T\right] + Q_k \tag{2.71}$$

Comparing Equation (2.71) to Equation (2.46), the equations are the same, however the linear state model allows the expectation value to be simplified.

Similar to the derivation of the EKF update equations, the uncertainty propagation equations can be derived by linearizing the state model

$$f(\mathbf{x}_k) = f(\hat{\mathbf{x}}_k) + F\Big|_{\mathbf{x}_k=\hat{\mathbf{x}}_k}(\mathbf{x}_k - \hat{\mathbf{x}}_k) \tag{2.72}$$

where the matrix F consists of the state model partials.

$$F = \frac{\partial f}{\partial \mathbf{x}}\Big|_{\mathbf{x}_k=\hat{\mathbf{x}}_k} \tag{2.73}$$

Substituting Equation (2.72) into Equation (2.71), the following expression is obtained.

$$P_{k+1|k} = E\left[\left(f(\hat{\mathbf{x}}_k) + F(\mathbf{x}_k - \hat{\mathbf{x}}_k) - \hat{\mathbf{x}}_{k+1|k}\right)\left(f(\hat{\mathbf{x}}_k) + F(\mathbf{x}_k - \hat{\mathbf{x}}_k) - \hat{\mathbf{x}}_{k+1|k}\right)^T\right] + Q_k \tag{2.74}$$

Simplifying the expression in Equation (2.74) using

$$\hat{\mathbf{x}}_{k+1|k} = f(\hat{\mathbf{x}}_k) \tag{2.75}$$

and Equation (2.52), Equation (2.46) is retained. Equations (2.75) and (2.46), which were formulated using the Jacobian linearization of the state model as shown in Equation (2.73), are the EKF uncertainty propagation equations.

2.7 Comparing Linear and Nonlinear Formulations

To compare the linear and nonlinear formulations of the Kalman filter, the purpose of each one must be understood. While each one suits the type of measurement model it is reflecting, both require the use of computing multi-dimensional expectation integrals with respect to a PDF. For a linear system, these integrals yield closed-form solutions. For a nonlinear system, however, an approximate solution usually must be formulated as the integrals often do not have a closed-form solution. So, while nonlinear Kalman update equations can be written, obtaining expressions for each of the covariance matrices is not always possible. Thus, the linearization method described is one of the approximate solutions used to obtain the required covariance matrices.

2.8 Kalman Filter Algorithm

Various approximate solution frameworks exist for the Kalman Filter for nonlinear systems. Sampling based approaches approximate the multi-dimensional expectation integrals as a sum of a finite number of weighted points within the domain. The points are computed by evaluating the integrand at specific locations. The difference in the solution frameworks is how the location and weights of these points are determined. For a general framework of selected points and given discretized time span, the steps of performing the Kalman Filter algorithm is as follows:

1. For a given Gaussian distribution, an associated state mean vector and covariance matrix are given. For all the frameworks, the locations of the points depend on the mean and covariance. So, the first step would be to generate the locations of the points,i.e. the state vector of each point.

2. Using the dynamical state model, propagate each point's state vector forward by one time step.

3. Compute the state mean and covariance of the propagated points , where N is the number of points. For the state covariance matrix, if applicable, the process noise Q can be included in the calculation.

$$\hat{\mathbf{x}}_{k+1|k} = \sum_{i=1}^{N} (w_i \mathbf{x_i}) \tag{2.76}$$

$$P_{x(k+1|k)} = \sum_{i=1}^{N} \left(w_i \left(\mathbf{x_i} - \hat{\mathbf{x}}_{k+1|k} \right) \left(\mathbf{x_i} - \hat{\mathbf{x}}_{k+1|k} \right)^T \right) + Q \tag{2.77}$$

4. Generate the locations of the new points using the propagated state mean and covariance. This is also called resampling.

5. Use the measurement model to generate the measurement vector for each propagated state point.

6. Using the newly generated measurement points, compute the measurement mean, measurement covariance and the cross-covariance. If applicable, the measurement noise covariance matrix R can be included in the computation of the measurement covariance.

$$\hat{\mathbf{y}}_{k+1} = \sum_{i=1}^{N} (w_i \mathbf{y_i}) \tag{2.78}$$

$$P_{y(k+1|k)} = \sum_{i=1}^{N} \left(w_i \left(\mathbf{y_i} - \hat{\mathbf{y}}_{k+1} \right) \left(\mathbf{y_i} - \hat{\mathbf{y}}_{k+1} \right)^T \right) + R \tag{2.79}$$

$$P_{xy(k+1|k)} = \sum_{i=1}^{N} \left(w_i \left(\mathbf{x_i} - \hat{\mathbf{x}}_{k+1|k} \right) \left(\mathbf{y_i} - \hat{\mathbf{y}}_{k+1} \right)^T \right) \tag{2.80}$$

7. Now that the covariance matrices have been computed, the general Kalman update equations can be applied to compute the posterior state mean and covariance.

This entire process can be iterated through the time span by simply using the posterior state mean and covariance generated at time step k as the initial mean and covariance at time step $k + 1$.

2.8.1 Sigma Point Filters

As mentioned in the last section, the different filter frameworks all share the similarity that the expectation integrals are approximated using a selected set of weighted sigma points. The difference is how these points and weights are specifically selected. The

general idea follows reproducing statistical moments of the assumed PDF. The moments can be mathematically constrained by first assuming that the expectation integral of a given function can be written as the finite summation of a number of weighted function values.

$$E\left[f(\mathbf{x})\right] \approx \sum_{i=1}^{N} w^i f(\mathbf{x}^i) \tag{2.81}$$

where N is the finite number of function values used for the approximation of the function, $\mathbf{x}^i$ are the sigma points, w^i represent the weights corresponding to each sigma point. A Taylor series approximation can be used to model the function up to a certain degree d, as shown below

$$f(\mathbf{x}) = f(\mathbf{x}^*) + \frac{\partial f(\mathbf{x}^*)}{\partial x_{N_1}} \delta x_{N_1} + \cdots + \frac{1}{d!} \frac{\partial^d f(\mathbf{x}^*)}{\partial x_{N_1} \ldots \partial x_{N_d}} \delta x_{N_1} \ldots \delta x_{N_d} \tag{2.82}$$

Substituting Equation (2.82) into Equation (2.81) and simplifying by factoring out constant terms from the expectation integrals and comparing the left and right-hand sides, the following set of equations can be obtained.

$$\begin{aligned} &\sum_{i=1}^{N} w^i = 1 \\ &\sum_{i=1}^{N} w^i \delta x_{N_1}^i = E\left[\delta x_{N_1}\right] \\ &\vdots \\ &\sum_{i=1}^{N} w^i \delta x_{N_1}^i \ldots \delta x_{N_d}^i = E\left[\delta x_{N_1} \ldots \delta x_{N_d}\right] \end{aligned} \tag{2.83}$$

These equations are known as the moment constraint equations (MCEs). Although there are an infinite number of moments for a given PDF, sigma points and weights often only need to be chosen to satisfy up to an order r. This means that if a given problem had $f(\mathbf{x})$ as a polynomial function with the highest degree $d \leq r$, then the sigma points and weights that can reproduce up to order d moments would provide an exact computation of the expectation integral. If $f(\mathbf{x})$ is not a polynomial function, then it would be expected due to the nature of Taylor series approximations that the more statistical moments that the sigma points can reproduce, the more accurate the computation will be for the expectation integral. The two most used quadrature schemes are the Smolyak quadrature scheme and the Unscented Transform (UT). The Smolyak quadrature scheme, although used frequently, is not attractive as the weights for the sigma points can be negative,

which could cause stability problems that could result in larger errors when computing the integrals.

2.8.2 Unscented Transform

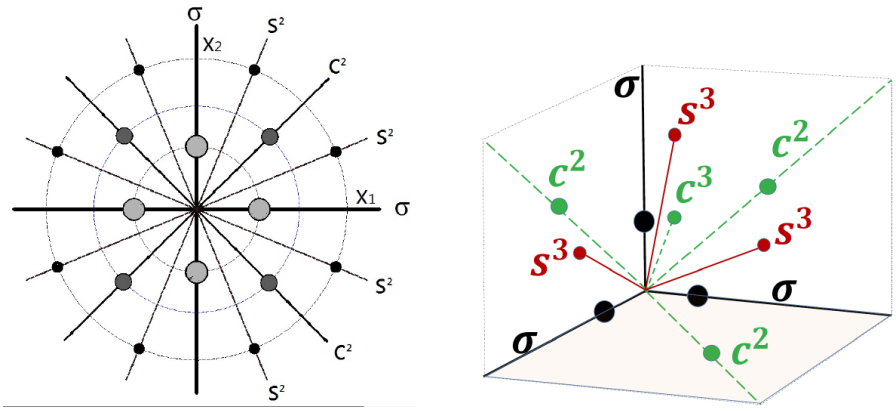
The UT selects points that are symmetric about the mean of the Gaussian PDF and lie on orthogonal axes defined by the columns of the covariance matrix. The UT points also only capture up to 3rd order statistical moments. Due to the symmetry, the odd moments are equal to 0 and already satisfied. So, only the even moments need to be satisfied in the selection of the sigma points and weights. For a given initial mean μ_0 and initial state covariance $(P_x)_0$, the sigma points and weights are generated using the following expressions

$$\begin{array}{rclcrcl} \mathbf{x}_0 & = & \boldsymbol{\mu}_0 & , & w_0 & = & \frac{\kappa}{n+\kappa} \\ \mathbf{x}_i & = & \boldsymbol{\mu}_0 + \sqrt{(n+\kappa)(P_x)_0} & , & w_i & = & \frac{1}{2(n+\kappa)} \\ \mathbf{x}_0 & = & \boldsymbol{\mu}_0 - \sqrt{(n+\kappa)(P_x)_0} & , & w_{i+n} & = & \frac{1}{2(n+\kappa)} \end{array} \tag{2.84}$$

where κ is a tuning parameter to adjust the location and weights of the points. Increasing κ will lead to an increase in distance from the mean and decrease the weights [8]. Implementing the UT version of the Kalman Filter is known as the Unscented Kalman Filter (UKF) [21]. The next section will discuss the CUT, which was developed to capture higher-order moments and hence provide more accurate computation of the expectation integral by using a higher-order Taylor series approximation.

2.8.3 Conjugate Unscented Transform

The primary aim of the CUT is to select points that continue to be symmetric about the mean, like the UT, but capture higher order moments, meaning they can satisfy the MCEs up to a larger degree d. The CUT incorporates more axes, called conjugate axes, to determine more optimal locations for the points to be placed such that higher order moments are captured as well as capturing cross moments. Figure 2.1 from [1] depicts the use of conjugate axes to construct a distribution of points such that higher order moments and cross moments are captured in n-D space, specifically 2-D and 3-D. The current set of CUT points generated go up to 8th order meaning they can capture up to 8th order moments. This also means that the maximum degree polynomial the CUT points generated can exactly evalute the expectation integral for is 8. Due to the symmetric nature of the points chosen, the odd moments all equal zero, so only the even



(a) CUT Points in 2D (b) CUT Points in 3D

Figure 2.1: Conjugate Axes for 2-D and 3-D CUT Point [1]

moments need to be considered for determining values of weights and scaling distances.

For a general $n \times 1$ state vector, CUT points up to 8th order have been pre-computed as long as $n \leq 6$. Using the pre-computed CUT points and the given initial mean and covariance, one can convert the CUT points to be centered around the state vector using

$$\mathbf{x_0} = \boldsymbol{\mu}_0 + \sqrt{(P_x)_0}\,(\mathbf{x_{CUT}}) \tag{2.85}$$

where the spread of the points around the state vector is determined by the state covariance matrix. The conversion process to a given mean and covariance is valid since the pre-computed CUT points were derived using a zero mean vector and identity covariance matrix. Equation (2.85) can be used as the framework for deciding the location of the points in the Kalman Filter algorithm and the entire algorithm can be implemented given the necessary information. This process is known as a CUT Filter. More details regarding the CUT derivation and methodology can be found in [1, 22–25]

2.8.4 Comparison of UKF and CUT

The benefits of implementing the CUT Filter as opposed to the UKF lies in data acquisition. For measurement data that is acquired at a high frequency, there is not significant data accuracy gain by implementing a CUT Filter over the UKF. As data measurements become more sparse, the CUT Filter performs better over time in terms of relative error compared to the UKF, at the cost of computational expense.

2.8.5 Air Traffic Tracking Example

To prove the claim about data acquisition, a general air traffic tracking problem can be posed. The equations of motion for the problem are given below.

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & \frac{\sin \Omega T}{\Omega} & 0 & -\frac{1-\cos \Omega T}{\Omega} & 0 \\ 0 & \cos \Omega T & 0 & -\sin \Omega T & 0 \\ 0 & \frac{1-\cos \Omega T}{\Omega} & 1 & \frac{\sin \Omega T}{\Omega} & 0 \\ 0 & \sin \Omega T & 0 & \cos \Omega T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \omega \tag{2.86}$$

where $\mathbf{x}_k$ is the state vector at time step k, Ω is the turn rate of the target, T is the time step, and ω is the state noise, which is characterized by the Gaussian PDF $\mathcal{N}(\omega : \mathbf{0}, Q)$. The state vector consists of $[x, v_x, y, v_y, \Omega]$, which provides the 2-D position and velocity in meters and meters per second respectively and the turn rate in radians per second. The state noise covariance Q is defined as

$$Q = L_1 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} & 0 \\ 0 & 0 & \frac{T^2}{2} & T & 0 \\ 0 & 0 & 0 & 0 & \frac{L_2}{L_1} T \end{bmatrix} \tag{2.87}$$

where $L_1 = 0.16$ and $L_2 = 0.01$. Furthermore, the target undergoes controlled trajectory changes: flying west for 125 seconds at 120 $\frac{m}{s}$, turning 90° towards south over a period of 90 seconds, flying south for 125 seconds at 120 $\frac{m}{s}$, turning 90° towards west over a period of 30 seconds, and finally flying west for 125 seconds at 120 $\frac{m}{s}$. Two stationary sensors are also used to make measurements on the target. More specifically, the measurement model is given by:

$$\mathbf{y}^j = \begin{bmatrix} \sqrt{(\mathbf{x}(1) - \mathbf{s}^j(1))^2 + (\mathbf{x}(2) - \mathbf{s}^j(2))^2} \\ \arctan\left(\frac{\mathbf{x}(2) - \mathbf{s}^j(2)}{\mathbf{x}(1) - \mathbf{s}^j(1)}\right) \end{bmatrix} \tag{2.88}$$

where $\mathbf{x}$ is the state vector of the object and $\mathbf{s}^j$ is the j^{th} sensor. The initial mean vector is given by $\boldsymbol{\mu}_0 = [25000, -120, 10000, 0, 1 \times 10^{-200}]^T$ and the initial covariance matrix is $(P_x)_0 = diag([1000^2, 10^2, 1000^2, 10^2, (\frac{\pi}{180})^2])$. For the simulation, the initial condition is randomly generated from a Gaussian distribution using the initial mean and

covariance and the time step used for all filters is 5 seconds. Given this information, one can implement the CUT Filter and the UKF to compare the accuracy of the methods. The standard deviation error bound plots for each method are shown in Figures 2.2-2.5.

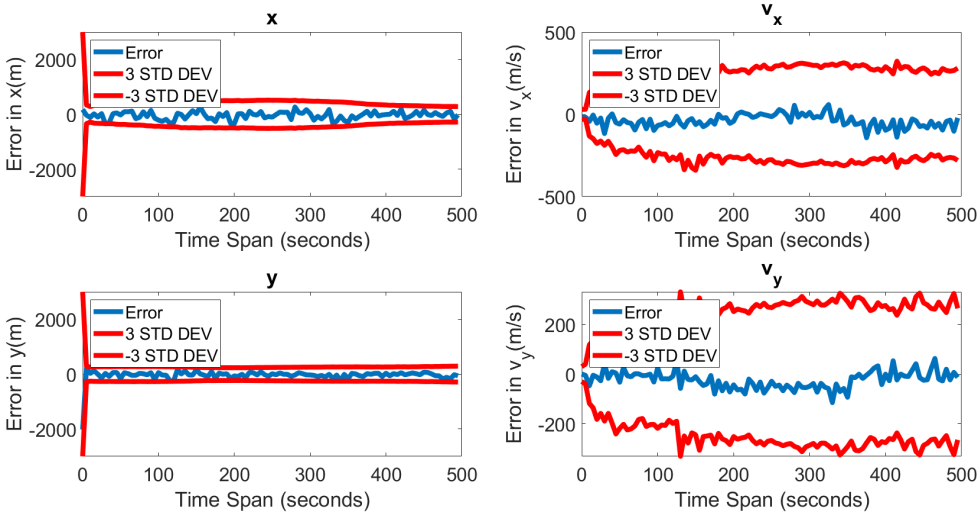


Figure 2.2: CUT8 State Error Performance

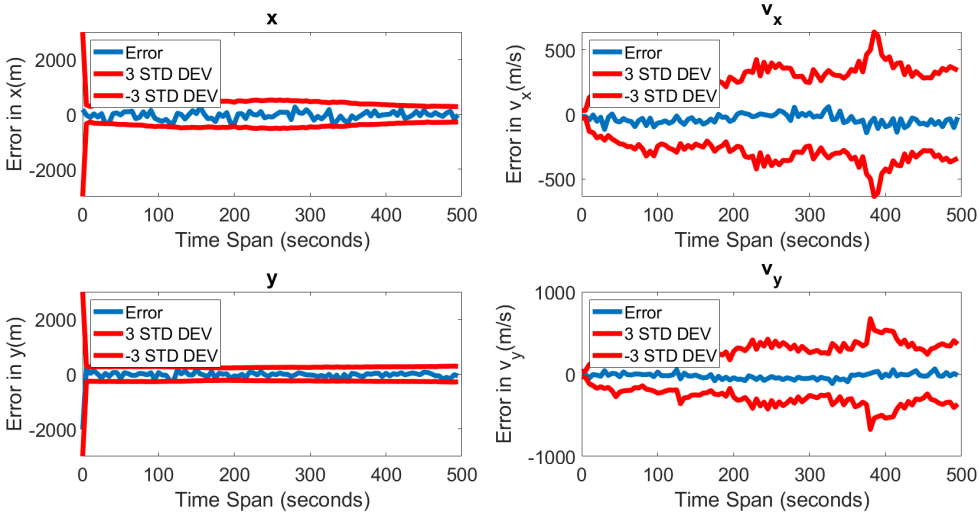


Figure 2.3: CUT6 State Error Performance

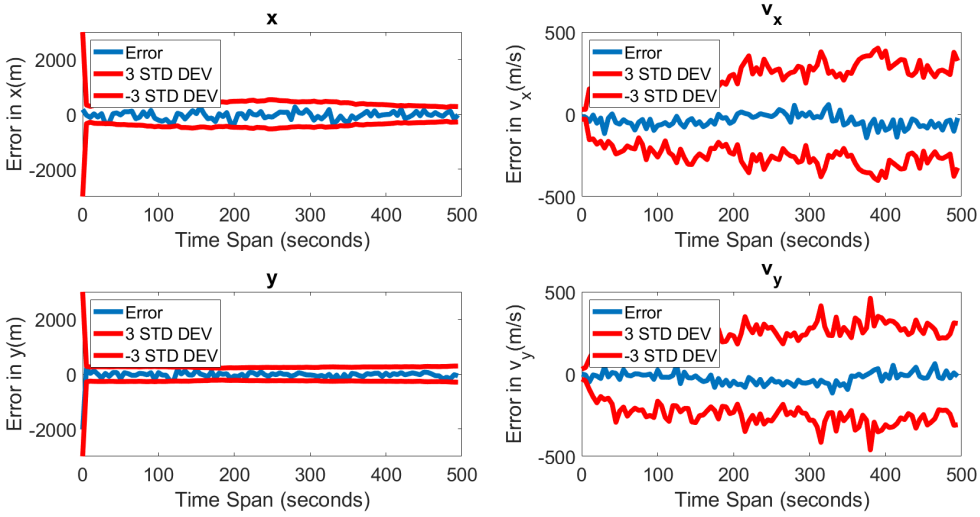


Figure 2.4: CUT4 State Error Performance

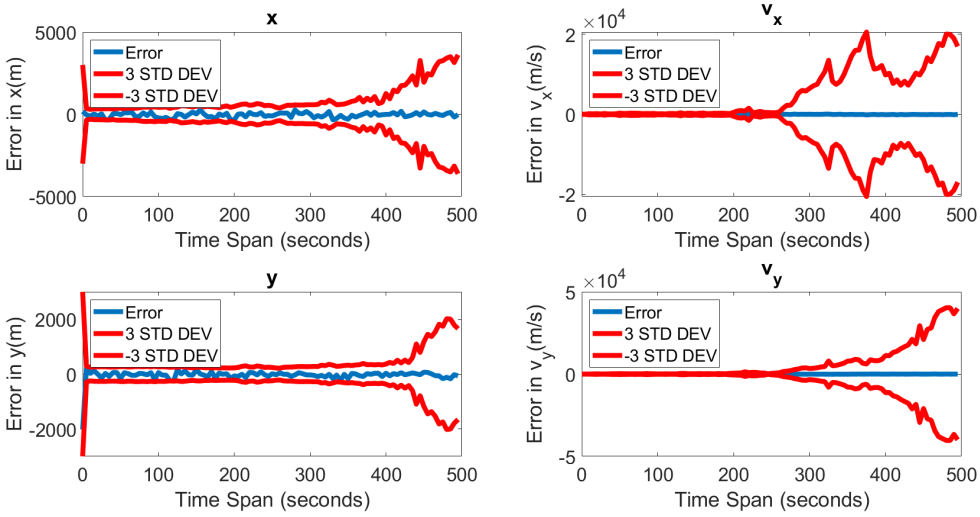


Figure 2.5: UKF State Error Performance

Only the error for the first four state variables are shown as the turn rate is manually changed at points in time, introducing an inherent error which will be prevalent throughout the time error plot. Also, following the times of the trajectory changes, the effect of the turn rate changing can be seen in the velocity error for the different filters. As can be seen from the axis limits, the higher-order filters perform better in terms of the error bounds and handling the changes. To further analyze the effects of the turn rate, the position estimate obtained from each filter can be superimposed compared to the true trajectory to visually analyze the performance of each filter, as shown in Figure 2.6. The

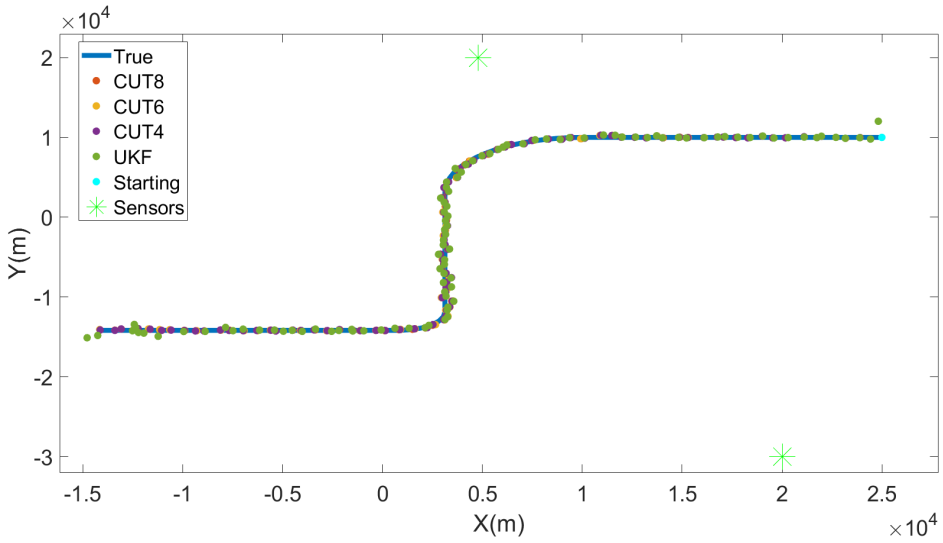


Figure 2.6: Filters' Position Estimate vs True Trajectory

position plot further shows that the UKF starts to diverge from the true trajectory by the end of the time span resulting in larger errors.

As the time between measurements gets larger, the CUT Filter performs better for accurately estimating the state. However, since the initial condition is generated randomly, one run might not exactly represent the capabilities of each filter for sparse data estimation. Thus, the performance of each of the four filters for 4 values of time steps is averaged over 100 runs to better represent each filter's capabilities. The root mean squared error (RMSE) plots are shown in Figures 2.7, 2.8, and 2.9 to showcase the improved performance.

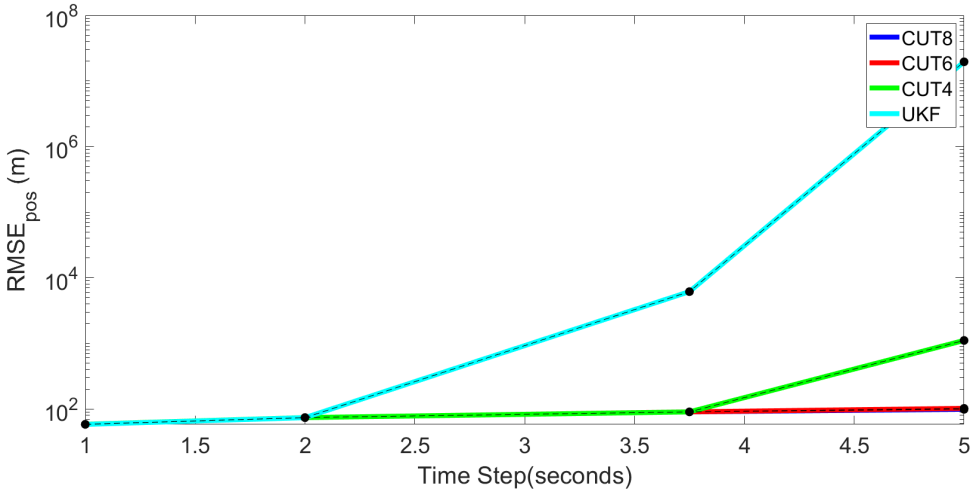


Figure 2.7: Position RMSE for Different Filters

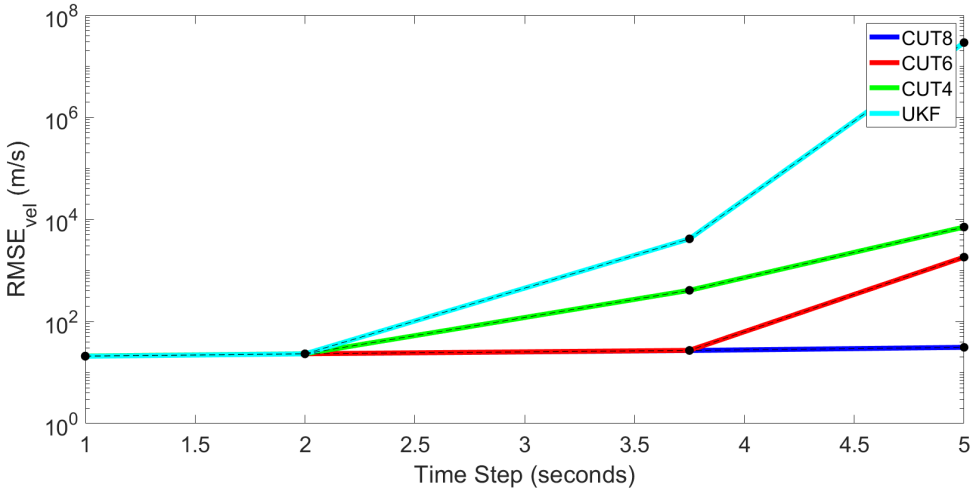


Figure 2.8: Velocity RMSE for Different Filters

Comparing the different filters, for a data acquisition time of 1 second, all the filters perform the same in tracking the position, velocity, and turn rate. However, as the time step between measurements increases, the CUT8 filter does the best at tracking followed by CUT6, CUT4, and UKF. For the 5 second case, even though CUT8 is more computationally expensive than the other filters, the error produced by the others makes those tracking algorithms completely unviable as the order of magnitude of the error is relative or greater than the order of magnitude of the state vector. This is important for cislunar space as several reasons can cause measurements to be infrequent or qualitatively poor, hence a CUT Filter will provide better implementation results than the UKF.

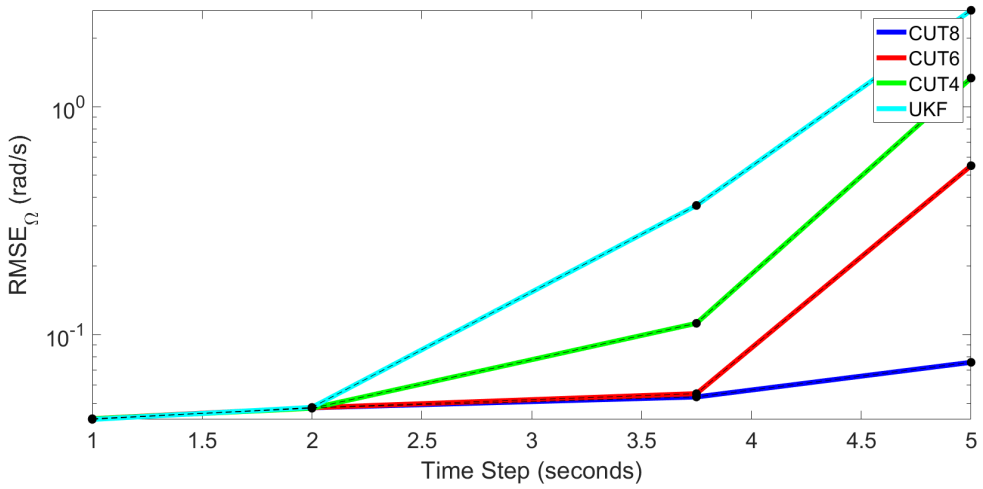


Figure 2.9: Turn Rate RMSE for Different Filters

2.9 Summary

This chapter outlined the general formulation of a Kalman Filter by first breaking down the general idea of state estimation and listing the required information to start state estimation. The maximum likelihood estimate was introduced to discuss the best, unbiased state estimator as well as set up the foundations of the state estimation problem. For a linear system, the maximum likelihood estimate was expanded upon by introducing bias in the form of a prior PDF weighting the likelihood function. This resulted in the linear Kalman Filter update equations. To confirm the propagation of a Gaussian PDF resulted in the posterior PDF remaining Gaussian, the CKE was used.

Next, the same problem was approached from a different route in minimizing the variance as opposed to maximizing the posterior PDF. The goal of this method was to provide a general set of Kalman update equations that did not rely on a linear measurement model.Using the result from the maximum a posteriori as a foundation, an expression for K was developed that allowed for the nonlinear Kalman update equations to be formulated.

Once the general update equations were derived, a linearization problem centered around the mean state vector was also presented to introduce the concept of approximate solutions for nonlinear systems in the form of the EKF update equations. Similar to the linear uncertainty propagation, the nonlinear uncertainty propagation was discussed. A brief comparison between the linear and nonlinear formulations was presented along with practicality constraints of the nonlinear method. The necessity of approximate

solutions was further established. For any given approximate solution framework, a general algorithm for the Kalman Filter was given and the key difference of generating the locations of points was stated. Then the general framework of the approximate solutions was discussed, which involves selecting sigma points to reproduce statistical moments of the PDF. The Smolyak quadrature scheme and UT were discussed, with more focus on the UT as it uses positive weights for the sigma points.

Finally, the CUT Filter was introduced as an approximate solution to the nonlinear Kalman Filter update equations and will be the filter used in the rest of this thesis. CUT points are generated to satisfy statistical moments up to a certain order while being symmetrically distributed and hold positive weights. For orders up to CUT8, the points have been pre-computed for up to 6-dimensional state models. These points can be converted to a given Gaussian PDF using the mean and covariance. Using those converted points, one can run through the Kalman Filter algorithm. For comparison, the UT and CUT were applied to an air traffic tracking problem. The results show clearly that for more sparse data, the CUT Filter does a better job in minimizing error.

Chapter 3
Sensor Tasking

3.1 Introduction to Mutual Information

The concept of mutual information (MI) relates to the information gain from observing a given set of random variables simultaneously. For example, given two random variables, the MI can be defined as the amount of information that can be gained about one random variable from the other. This concept can be applied to tracking targets in a sensor tasking problem. A typical tasking problem consists of using multiple sensors to optimally track a group of targets, where optimality is defined by maximum reduction of system uncertainty. In order to reduce the system uncertainty, the sensor's ability to observe a target, or the sensor performance, has to be quantified. Using the concept of MI, the sensor network can be scheduled ahead of time to track targets in such a way that over time, the individual uncertainty of each target will be reduced. In this regard, MI can be defined as reducing system uncertainty, and maximizing the MI of a given system will minimize the uncertainty.

The following sections will first discuss the mutual information connection to the previous chapter's topic of filtering. Then, the MI will be considered as a sensor performance metric and compared to other performance metrics, specifically the Fisher Information Matrix (FIM) [26]. The general derivation of the mutual information expression will be provided for a singular time step. Since MI can be extended to multiple time steps, the formulation of the exhaustive search algorithm will be shown. As the exhaustive search is computationally expensive for larger state and measurement models or for several time steps, a sub-optimal solution is provided that limits the domain of possible schedules to each time step iteration. The idea is expanded to a receding horizon window approach. An example is shown for tracking targets around the Earth and in cislunar space.

3.1.1 Connecting CUT Filter to Mutual Information

As discussed in the previous chapter, the CUT Filter relates to using measurements to estimate the state vector at a given time step. However, the filter assumes measurements are readily available. For tracking one particular target, one can dynamically reorient the sensor to track the target as soon as it is visible to reduce the uncertainty in the target's state vector. However, using one sensor to track one target is not an efficient method of resource allocation. The number of space targets far exceeds the number of sensors available for tracking, so the problem arises of tracking multiple targets with a select few sensors. This means that unless some of the targets follow similar trajectories and can be captured in the same field of view, at a given time step, not all targets will be observed. Taking this into consideration, one can employ the concept of mutual information to generate the optimal sensor tasking schedule for a given number of sensors and targets such that over time, the uncertainty of all targets will decrease.

3.1.2 Sensor Performance Metric

The foundation of mutual information applied to a target tracking application stems from being a sensor performance metric. For the general optimization problem, the cost function is a function of the states of all targets at all time steps as well as all possible measurements from all sensors, or $J = J(\mathbf{X}_{(0:N_{TS}-1)}\mathbf{Y}_{(0:N_{TS}-1)})$, where J is the cost function and N_{TS} is the total number of time steps. In regards to the optimization problem, the cost function is also defined as a sensor performance metric and is highly problem dependent. The choice of sensor performance can be based on several factors, such as measurement model type, estimation algorithm, and filtering framework chosen. For example, for a linear measurement model, the FIM can be used as the inverse provides the Cramer-Rao lower bound (CRLB), which is the minimum state error covariance matrix that can be achieved. Since the measurement model is linear, the CRLB is guaranteed. For nonlinear systems, the FIM can be used iff the linearized model of the nonlinear measurement model is reasonably accurate. This is important since the FIM for a nonlinear system is only an approximation that comes from the Taylor series expansion of the Kullback-Leibler (K-L) Divergence [27, 28]. If the approximation is reasonably accurate, then the FIM is a viable sensor performance metric, otherwise it is infeasible. The mutual information, however, is a function of the entire pdf and is not an approximation like the FIM.

3.1.3 Kullback-Leibler Divergence

The K-L Divergence is a quantitative measure of the difference between two pdfs, sometimes referred to as the relative entropy of a system [29]. Taking an example for a random variable r with two events H_1 and H_2 each with a respective pdf $f_1(r)$ and $f_2(r)$, the prior and posterior probability of H_i can be defined as $p(H_i)$ and $p(H_i|r = x)$, respectively, where $i = 1, 2$ and x is the observation made for r. The posterior probability of each event can be defined as follows:

$$p(H_i|r = x) = \frac{p(H_i)f_i(r)}{p(H_1)f_1(r) + p(H_2)f_2(r)} \tag{3.1}$$

The relation between the two pdfs and the event probabilities can be expressed as shown.

$$\log \frac{f_1(r)}{f_2(r)} = \log \frac{p(H_1|r = x)}{p(H_2|r = x)} - \log \frac{p(H_1)}{p(H_2)} \tag{3.2}$$

The left-hand-side of Equation (3.2) represents the information gained making the observation $r = x$ in favor of event H_1 compared to event H_2. Using this, an expression for the mean information favoring H_1 in terms of H_2 can be obtained.

$$I(1 : 2) = \int f_1(r) \log \frac{f_1(r)}{f_2(r)} dr \tag{3.3}$$

Similarly, the mean information favoring H_2 in terms of H_1 can be expressed as shown.

$$I(2 : 1) = \int f_2(r) \log \frac{f_2(r)}{f_1(r)} dr \tag{3.4}$$

Finally, a definition for the divergence can be written as the sum of the mean information expressions.

$$D_{KL}(1, 2) = I(1 : 2) + I(2 : 1) = \int (f_1(r) - f_2(r)) \log \frac{f_1(r)}{f_2(r)} dr \tag{3.5}$$

Substituting Equation (3.2) into Equation (3.5), the following expression is obtained.

$$D_{KL}(1, 2) = \int f_1(r) \log \frac{p(H_1|r)}{p(H_2|r)} - \int f_2(r) \log \frac{p(H_1|r)}{p(H_2|r)} dr \tag{3.6}$$

Note that the logarithms corresponding to the prior probabilities are ignored. The expression shown in Equation (3.6) is representative of the divergence between H_1 and

H_2 and is known as the K-L Divergence. The mutual information can be considered as the expected value of the K-L Divergence of the posterior pdf from the prior pdf. So maximizing the mutual information of a given system would mean the posterior pdf of the system achieves the maximum divergence from the prior pdf. Since the covariance update equation uses measurements to decrease the state covariance matrix, the maximum divergence would result in a minimization of the posterior pdf, which is equivalent to minimizing the system uncertainty.

3.2 Derivation of Mutual Information Expression

Given two random vectors $\mathbf{x}$ and $\mathbf{y}$, one can write the MI expression between both as

$$I(\mathbf{x};\mathbf{y}) = \int\int p(\mathbf{x},\mathbf{y}) \ln\left(\frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}\right) d\mathbf{x}d\mathbf{y} \tag{3.7}$$

For application to state estimation, $\mathbf{x}$ can be considered to be the state vector and $\mathbf{y}$ is the measurement vector. As used in Chapter 2, Bayes' theorem can be used to modify Equation (3.7). The previous Bayes' theorem can also be written as

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{y})} \tag{3.8}$$

Substituting Equation (3.8) into Equation (3.7), the following equation is obtained

$$I(\mathbf{x};\mathbf{y}) = \int\int p(\mathbf{x},\mathbf{y}) \ln\left(\frac{p(\mathbf{x}|\mathbf{y})}{p(\mathbf{x})}\right) d\mathbf{x}d\mathbf{y} \tag{3.9}$$

Further analyzing Equation (3.9), it can be noted that the expression can be interpreted as the expected value with respect to the measurement vector of the K-L divergence of the posterior pdf from the prior pdf as mentioned previously. From this, it is obvious that the more the posterior pdf is different, or diverges, from the prior pdf, the more information that is gained about the system from certain measurement-state pairings, which in term leads to less uncertainty in the system.

$$\int\int p(\mathbf{x},\mathbf{y}) \ln\left(\frac{p(\mathbf{x}|\mathbf{y})}{p(\mathbf{x})}\right) d\mathbf{x}d\mathbf{y} = E_{\mathbf{y}}[D_{KL}(p(\mathbf{x}|\mathbf{y})||p(\mathbf{x}))] \tag{3.10}$$

Although the approach can be applied to linear systems, the initial motivation was to provide a sensor performance metric that is feasible for nonlinear systems. As discussed

in the last chapter, several nonlinear systems do not yield closed-form analytic solutions and hence the expectation integrals must be evaluated numerically. Assuming the state can be modeled accurately by a Gaussian pdf, the following can be defined

$$p(\mathbf{x}, \mathbf{y}) \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{x} \\ \mathbf{y}\end{bmatrix} : \begin{bmatrix}\boldsymbol{\mu_x} \\ \boldsymbol{\mu_y}\end{bmatrix}, P\right) \tag{3.11}$$

where P is the joint covariance matrix of the states and measurements. The matrix P can be broken down into the state covariance, measurement covariance, and cross-covariance matrices.

$$P = \begin{bmatrix} P_x & P_{xy} \\ P_{xy}^T & P_y \end{bmatrix} \tag{3.12}$$

Applying this to the MI expression, the following expression is obtained.

$$I(\mathbf{x}; \mathbf{y}) = \frac{1}{2} \ln\left(\frac{|P_x||P_y|}{|P|}\right) \tag{3.13}$$

The determinant of the joint covariance matrix P requires using the formula for determinants of block matrices, which is acceptable since the state and measurement covariance matrices are invertible. Since the formula has two forms for the determinant, the following two expressions are obtained.

$$I(\mathbf{x}; \mathbf{y}) = \frac{1}{2} \ln\left(\frac{|P_x|}{|P_x - P_{xy}P_y^{-1}P_{xy}^T|}\right) \tag{3.14}$$

$$I(\mathbf{x}; \mathbf{y}) = \frac{1}{2} \ln\left(\frac{|P_y|}{|P_y - P_{xy}^T P_x^{-1} P_{xy}|}\right) \tag{3.15}$$

Both are valid expressions for the MI, however Equation (3.14) compares the prior state covariance in the numerator to the posterior state covariance in the denominator, which is computed using the Kalman covariance update equation, whereas Equation (3.15) maximizes the likelihood of making a measurement by reducing the measurement covariance. For any time step k, the MI at time step $k+1$ can be computed using the following expressions

$$I(\mathbf{x}; \mathbf{y}) = \frac{1}{2} \ln\left(\frac{|P_{x(k+1|k)}|}{|P_{x(k+1|k)} - P_{xy(k+1|k)}P_{y(k+1|k)}^{-1}P_{xy(k+1|k)}^T|}\right) \tag{3.16}$$

$$I(\mathbf{x};\mathbf{y}) = \frac{1}{2}\ln\left(\frac{|P_{y(k+1|k)}|}{P_{y(k+1|k)} - P^T_{xy(k+1|k)}P^{-1}_{x(k+1|k)}P_{xy(k+1|k)}|}\right) \tag{3.17}$$

For a given number of targets and sensors, these MI expressions can be used to determine at each time step which sensors should look at which targets and the entire process can be iterated through the time span.

3.3 Exhaustive Search

The exhaustive search algorithm consists of searching the entire domain of all time steps, targets, and sensors for the optimal sensor tasking schedule. Starting from first principles, several assumptions have been made to generate the MI expression. However, there is still freedom for using this expression, specifically in the definition of the state vector $\mathbf{x}$ and the measurement vector $\mathbf{y}$. Instead of taking the state vector of a target at one time step and iterating through the time span, the state vector could consist of the target's state vectors at all points in the given time span concatenated into one joint state vector. The same concept can be applied to the measurement vector. To alleviate the computational expense, the formulation for the Gaussian pdf will remain the same for the state and measurement vector, meaning the joint vector of all states and measurements for all targets will be approximated using a Gaussian pdf. For space-based targets, the process noise is many times negligibly small and does not need to be considered. However, measurement noise does need to be considered as several applications involve tracking targets over large distances, which increase the chance of background noise diluting measurements.

3.3.1 Formulation of Joint Mutual Information

For the joint state and measurement vector, the pdf in Equation (3.11) turns into

$$p\left(\mathbf{X}^{1:N_T}_{0:N_{TS}-1}, \mathbf{Y}^{(1:N_T,1:N_S)}_{0:N_{TS}-1}\right) \tag{3.18}$$

where N_T represents the total number of targets and N_S represents the total number of sensors. To compute the joint mutual information, the pdf in Equation (3.18) must be used to derive the MI expression. However, the MI expression was derived generally, so simply plugging in the joint state and measurement vectors will suffice. For this thesis, the data association problem for measurements made on multiple targets will be assumed to be solved, meaning the measurements made on each target can be resolved

from each other. This assumption helps simplify the joint MI problem because instead of considering all targets at once, the joint MI problem can be reduced to considering each target individually and summing the MI of each target.

$$I\left(\mathbf{X}_{0:N_{TS}-1}^{1:N_T};\mathbf{Y}_{0:N_{TS}-1}^{(1:N_T,1:N_S)}\right)=\sum_{i=1}^{N_T} I\left(\mathbf{X}_{0:N_{TS}-1}^{(i)};\mathbf{Y}_{0:N_{TS}-1}^{(i,1:N_S)}\right) \tag{3.19}$$

Applying this to the MI expressions previously developed in Equations (3.14) and (3.15), the joint mutual information expression for the ith target can be formulated as shown in the two following expressions

$$I\left(\mathbf{X}_{0:N_{TS}-1}^{(i)};\mathbf{Y}_{0:N_{TS}-1}^{(i,1:N_S)}\right)= \frac{1}{2}\ln\left(\frac{\left|S_{(0:N_{TS}-1)}^{(i)}\right|}{\left|S_{(0:N_{TS}-1)}^{(i)}-\Gamma_{(0:N_{TS}-1)}^{(i,1:N_S)}\left(\Xi_{0:N_{TS}-1}^{(i,1:N_S)}\right)^{-1}\left(\Gamma_{(0:N_{TS}-1)}^{(i,1:N_S)}\right)^{T}\right|}\right) \tag{3.20}$$

$$I\left(\mathbf{X}_{0:N_{TS}-1}^{(i)};\mathbf{Y}_{0:N_{TS}-1}^{(i,1:N_S)}\right)= \frac{1}{2}\ln\left(\frac{\left|\Xi_{(0:N_{TS}-1)}^{(i,1:N_S)}\right|}{\left|\Xi_{(0:N_{TS}-1)}^{(i,1:N_S)}-\left(\Gamma_{(0:N_{TS}-1)}^{(i,1:N_S)}\right)^{T}\left(S_{0:N_{TS}-1}^{(i)}\right)^{-1}\Gamma_{(0:N_{TS}-1)}^{(i,1:N_S)}\right|}\right) \tag{3.21}$$

where $S_{(0:N_{TS}-1)}^{(i)}$ is the state covariance matrix for all states of the ith target at all time steps, $\Xi_{(0:N_{TS}-1)}^{(i,1:N_S)}$ is the measurement covariance matrix for the measurements made by all sensors of the ith target at all time steps, and $\Gamma_{(0:N_{TS}-1)}^{(i,1:N_S)}$ is the cross-covariance of all the states of the ith target with measurements made by all the sensors at all time steps.

To compute the joint mutual information given an initial mean estimate, with size $n\times 1$, and covariance, with size $n\times n$, for the state of each target, one can implement the use of sigma points to numerically compute the required covariance matrices. Using the initial mean and covariance, the sigma points can be generated and propagated through the entire time span using the dynamical model with state vectors being recorded at each time step. For each sigma point, all the state vectors can be stacked as one column vector of length $n\times N_{TS}$.

Similarly, the measurement model can be used to make measurements of each state at each time step with each sensor. For the concatenation process, the measurement vectors are first stacked by all sensors at each time step and then by all time steps. So if given N_S sensors each with measurement vector of size $m\times 1$, the joint measurement vector

would be of length $m \times N_{TS} \times N_S$. Once this process is extended to all sigma points, the following summation expressions can be used to generate the respective covariance matrices that are part of the joint MI expression.

$$\hat{\mathbf{X}} = \sum_{b=1}^{N} (w_b \mathbf{X_b}) \tag{3.22}$$

$$S = \sum_{b=1}^{N} \left(w_b \left(\mathbf{X_b} - \hat{\mathbf{X}} \right) \left(\mathbf{X_b} - \hat{\mathbf{X}} \right)^T \right) \tag{3.23}$$

$$\hat{\mathbf{Y}} = \sum_{b=1}^{N} (w_b \mathbf{Y_b}) \tag{3.24}$$

$$\Xi = \sum_{b=1}^{N} \left(w_b \left(\mathbf{Y_b} - \hat{\mathbf{Y}} \right) \left(\mathbf{Y_b} - \hat{\mathbf{Y}} \right)^T \right) + R \tag{3.25}$$

$$\Gamma = \sum_{b=1}^{N} \left(w_b \left(\mathbf{X_b} - \hat{\mathbf{X}} \right) \left(\mathbf{Y_b} - \hat{\mathbf{Y}} \right)^T \right) \tag{3.26}$$

Note that for the state covariance matrix sigma point computation, the process noise matrix was ignored as mentioned previously, however, to include process noise, simply adding the process noise covariance matrix to the state covariance matrix can incorporate the process noise. The process noise covariance matrix would simply need to be constructed as a joint noise covariance matrix for all time steps.

Since the covariance matrices were constructed using the measurements of each sensor for each target, the matrices are "over-constructed," meaning since each sensor is not guaranteed to see all targets at all time steps, when computing the MI value, only the respective rows and columns corresponding to a given sensor observing a target need to be considered and the other rows and columns can be removed so that they don't impact the MI calculation.

3.3.2 Maximizing Mutual Information

Now that the mutual information for a given set of measurements can be computed, the next step is to determine, for the entire span, which sensors should look at which targets and when. More generally, the sensor parameters primarily dictate the value of the MI expression since that is the controllable part of the overall formula, meaning for a given sensor schedule the only parts of the MI expression that will change are the joint cross-covariance and the joint measurement covariance. Thus, optimizing the sensor parameters such that the MI value is maximized is the main objective.

For this thesis, the sensor parameters will be restricted to the pointing angle of the sensors. Several factors can affect the pointing angle, however, such as the detection probability of a target by a given sensor, which will depend on the field-of-view (FOV) and the distance from the sensor to the target. In this case, optimizing sensor parameters means determining which targets should be observed by which sensors at each time step. Looking at a simple example of 2 sensors, 4 targets, and 3 time steps, the number of different possibilities of sensors looking at targets, assuming each sensor has to observe an object at each time step, is $4^{2 \cdot 3} = 4096$. Generalizing this equation, the number of different possibilities is $(N_T)^{N_S * N_{TS}}$. For a low dimensional system, the number of possibilities will not result in large computational expense and can be solved using the exhaustive search algorithm. However, as can be seen from the general equation, increasing the number of time steps will drastically increase the number of possibilities as the exponent changes. For 10 time steps, the previous example results in over 1 trillion possibilities, which is computationally infeasible. This means a sub-optimal method is required to reduce computational costs while also providing a sensor schedule that can be utilized to overall minimize the uncertainty in a system.

The next section will discuss introducing decision variables such that the problem of searching an entire domain for the optimal sensor schedule can be transformed into a rational function, which can be input into numerical solvers like GloptiPoly to obtain the maximum solution while considering semi-definite programming (SDP) relaxations to optimize the searching process. Note that using GloptiPoly does not reduce the total number of possibilities but simply provides an efficient method for searching the entire domain, which will be more useful for the sub-optimal methodology than for the exhaustive search.

3.3.3 Decision Variables

As mentioned in Section 3.3.1, the covariance matrices are "over-constructed" and must be reduced for a given sensor schedule, meaning certain rows and columns must be removed based off whether a certain sensor is observing a certain target. This problem can be defined as a binary integer optimization problem, where the decisions consist of either including sensor measurements of a target at a given time step or not. The MI expression defined in Equation (3.19) can be turned into a maximization problem

$$\max_{\alpha_k^{(i,j')}} : \sum_{i=1}^{N_T} I\left(\mathbf{X}^{(i)}_{(0:N_{TS}-1)}; \alpha^{(i,1:N'_S)}_{(0:N_{TS}-1)} \mathbf{Y}^{(i,1:N'_S)}_{(0:N_{TS}-1)}\right) \tag{3.27}$$

where $\alpha_k^{(i,j')}$ represents the decision to have the j' sensor observing the ith object at time step k. The main difference between Equation (3.19) and (3.27) is that now there are considered to be N_S' sensors instead of N_S sensors. The key difference is that instead of considering each sensor as having multiple pointing angles as operating modes, now each sensor at each of those operating modes is considered to be its own "sensor". For example, if there are 2 sensors and 3 targets, each of the sensors can observe each of the targets, assuming the targets are not blocked, meaning each of the sensors have 3 operating modes. So, N_S would equal 2 since there are only two physical sensors, but N_S' would be $2 \times 3 = 6$ to represent each operating mode. For the problem setup so far, this is more of a notation choice than re-structuring the framework.

To incorporate the decision variables into the MI calculation, one can introduce the decision variables into the joint measurement and cross-covariance matrices computed. Similarly, the decision variables can be introduced directly into the concatenated measurement vectors of each sigma point. This can be achieved by multiplying the measurement vector of the ith target by the jth sensor at time step k by $\alpha_k^{(i,j')}$. For a given target and sensor at a given time step, $\alpha_k^{(i,j')}$ turns into $\alpha_k^{(i,j)}$ since the number of operating modes are dictated by the targets.

Once the decision variables are introduced into the covariance matrices, some mathematical precautions need to be enforced. Specifically in the measurement covariance matrix, introducing the decision variables will lead to the multiplication of two separate decision variables. For the general case of two decision variables $\alpha_{k_1}^{(i,j_1')}$ and $\alpha_{k_2}^{(i,j_2')}$, if $k_1 = k_2$ and $j_1' = j_2'$, then $\alpha_{k_1}^{(i,j_1')}\alpha_{k_2}^{(i,j_2')} = 1$. This ensures that the determinant does not equal 0 for any row and column deletion due to sensor removal.

Now, the MI expression, specifically the version in Equation (3.21), can be rewritten using the optimization problem notation as

$$I : \max_{\boldsymbol{\alpha}} \sum_{i=1}^{N_T} \left(\frac{\left|\Xi^{(i)}\right|}{\left|\Xi^{(i)} - \left(\Gamma^{(i)}\right)^T \left(S^{(i)}\right)^{-1} \Gamma^{(i)}\right|} \right) = \sum_{i=1}^{N_T} \left(\frac{p_i(\boldsymbol{\alpha})}{q_i(\boldsymbol{\alpha})} \right) = \frac{p(\boldsymbol{\alpha})}{q(\boldsymbol{\alpha})} \tag{3.28}$$

Note that the $\frac{1}{2}$ and the natural logarithm included in Equation (3.21) were discarded as the overall optimality does not suffer and a rational function now lends itself to being used in a binary-integer programming solvers. To constrain the values of the decision variables to 0 or 1, the following constraint is enforced.

$$\alpha_k^{(i,j')} - \left(\alpha_k^{(i,j')}\right)^2 = 0 \tag{3.29}$$

Recalling the 2 sensors and 3 targets example and considering 2 time steps, the decision variables can be listed out.

$$\begin{bmatrix} \alpha_1^{(1,1)} & \alpha_1^{(2,1)} & \alpha_1^{(3,1)} & \alpha_1^{(1,2)} & \alpha_1^{(2,2)} & \alpha_1^{(3,2)} \\ \alpha_2^{(1,1)} & \alpha_2^{(2,1)} & \alpha_2^{(3,1)} & \alpha_2^{(1,2)} & \alpha_2^{(2,2)} & \alpha_2^{(3,2)} \end{bmatrix} \tag{3.30}$$

From the matrix shown, N_S' can be understood to be the length of each row, which represents the number of variables at each time step. In the first row, the first three decision variables represent the first sensor looking at each of the three targets at the first time step. Similar logic can be applied to the rest of the decision variables to multiply them by the correct measurement vectors.

Using this framework, once the polynomial expressions for the numerator and denominator are determined, constraint equation (3.29) along with other problem-dependent constraint equations can be formulated and passed through to a numerical solver to optimize the sensor schedule for the given time span. As discussed before, the computational expense remains low for low-dimensional problems. However, for higher dimensional problems, a sub-optimal method must be implemented.

3.4 Sequential Computation

For a given tasking problem, the formulation of the exhaustive search measurement covariance matrix depends on the number of states in the measurement model, the number of sensors, and the number of time steps. For a simple case of a 3-state measurement model for 2 sensors for 2 time steps, the resultant covariance matrix would be of size 12×12. Incorporating the decision variables, the problem turns into taking a determinant of a 12×12 symbolic matrix. Although there are properties of the covariance matrix that would speed up this computation time, it is still quite expensive for any realistic scenario and gets worse once more time steps or sensors are introduced. For this reason, a sub-optimal methodology can be utilized to reduce the computational expense. An important note is that even though the sub-optimal method still produces a sensor schedule that given a long enough time span will reduce the uncertainties of all targets, the sub-optimality relates to maximizing a different cost function for each iteration that will not produce the globally optimal solution that the exhaustive search algorithm will provide. However, since the MI is nonnegative, nondecreasing, and submodular, the property of diminishing returns can be utilized to obtain a sub-optimal solution that is guaranteed to be within a bound of $1 - 1/e$ of the globally optimal solution. [20]

3.4.1 Sequential-in-Time

The sequential-in-time method performs the optimization problem at a given time step. Starting at the initial time step, the problem is formulated for a singular time step, meaning the numerator and denominator of the mutual information expression are found for the first time step. Next, the decision variables are solved for to maximize the mutual information, while the decision variables at subsequent time steps are set as zero. For an example of 2 targets, 2 sensors, and 3 time steps, the matrix shown shows the decision variables for one of the targets.

$$\begin{bmatrix} \alpha_1^{(1,1)} & \alpha_1^{(1,2)} & \alpha_1^{(2,1)} & \alpha_1^{(2,2)} \\ \alpha_2^{(1,1)} & \alpha_2^{(1,2)} & \alpha_2^{(2,1)} & \alpha_2^{(2,2)} \\ \alpha_3^{(1,1)} & \alpha_3^{(1,2)} & \alpha_3^{(2,1)} & \alpha_3^{(2,2)} \end{bmatrix} \tag{3.31}$$

Applying the sequential-in-time methodology, for the first iteration, Equation (3.31) would become

$$\begin{bmatrix} \alpha_1^{(1,1)} & \alpha_1^{(1,2)} & \alpha_1^{(2,1)} & \alpha_1^{(2,2)} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.32}$$

Once the initial time step is optimized, the next time step is solved for by first obtaining the numerator and denominator for the first two time steps. The decision variables optimized at the first time step are considered to be constant and substituted into Equation (3.32). So, only the decision variables at the current time step are being optimized again An example of solved decision variables is shown.

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ \alpha_2^{(1,1)} & \alpha_2^{(1,2)} & \alpha_2^{(2,1)} & \alpha_2^{(2,2)} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.33}$$

This process can be iterated through the time span until all decision variables from the exhaustive search optimization problem are determined.

3.4.2 Receding Horizon Window

For the sequential-in-time algorithm, at each iteration, one solves for the decision variables by formulating the MI expression at the current time step and passing the function to an optimization solver. However, one could instead formulate a filtering scheme similar

to the CUT Filter that tests all possibilities at each time step to maximize mutual information and iterate accordingly. For a given number of targets and a select few sensors at one time step, the possibilities remain relatively low and computationally feasible, especially if parallel processing is utilized. For this reason, instead of following a sequential-in-time approach, a receding horizon window can be applied. This approach is a sub-optimal method that extends the sequential-in-time method closer to the exhaustive search algorithm. The primary difference is while the sequential-in-time method optimizes the decision variables at one time step, the receding horizon window approach optimizes the decision variables for a finite number of time steps that do not take up the whole time span. An example would be optimizing the next 5 time steps of a time span consisting of 200 time steps. The objective of this approach is to strike a balance between increasing optimality and getting closer to global optimal solution and alleviating computational expense.

3.5 Numerical Results

The following examples will demonstrate the receding horizon window approach to a mutual information based sensor tasking algorithm.

3.5.1 Example 1: Earth Target Tracking

Consider 5 targets in orbits around the Earth governed by 3-D two-body dynamics, where the equations of motion are as follows:

$$\ddot{\mathbf{r}} = - \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ \frac{GM}{d^3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{GM}{d^3} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{GM}{d^3} & 0 & 0 & 0 \end{bmatrix} \mathbf{r} \tag{3.34}$$

where $\mathbf{r}$ is the state vector $[x, y, z, v_x, v_y, v_z]$ representing the position and velocity of each target, d is distance from the Earth to each target, G is the universal gravitational constant expressed in kilograms, kilometers, and seconds, and M is the total mass of each respective Earth-target system.

3.5.1.1 Initial Conditions

Table 3.1 shows each targets' orbital elements: semi-major axis a, eccentricity e, inclination i, right ascension of the ascending node Ω, argument of periapsis ω, and true anomaly θ. The orbital elements were used to compute the initial state vector consisting of the

	Target 1	Target 2	Target 3	Target 4	Target 5
a (km)	6925	6925	6925	6707.5	6925
e	1.381×10^{-4}	1.402×10^{-4}	1.35×10^{-4}	6.305×10^{-4}	1.589×10^{-4}
i (deg)	53.05	53.05	53.05	53.01	53.06
Ω (deg)	267.4028	267.799	267.2504	126.1677	267.1687
ω (deg)	89.6372	99.2824	91.3349	94.8006	115.197
θ (deg)	270.479	260.8327	268.7797	277.2313	244.9185

Table 3.1: Targets' Orbital Elements

position and velocity for each target, which was used as the initial mean vector $\boldsymbol{\mu}_0$. The initial state covariance $(P_x)_0$ follows the general covariance matrix shown below

$$\begin{bmatrix} x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & x^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & x^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & (\frac{x}{100})^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (\frac{x}{100})^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (\frac{x}{100})^2 \end{bmatrix} \tag{3.35}$$

where x for each respective target is $[10, 11.25, 12.5, 13.75, 15]$. As mentioned before, no state noise was included as it can generally be neglected for space systems.

The measurement model for the i^{th} target and the j^{th} sensor consists of three values: range, elevation, and azimuth, calculated as shown below.

$$\mathbf{y}^{(i,j)} = \begin{bmatrix} \sqrt{(\mathbf{r}^i(1) - \mathbf{s}^j(1))^2 + (\mathbf{r}^i(2) - \mathbf{s}^j(2))^2 + (\mathbf{r}^i(3) - \mathbf{s}^j(3))^2} \\ \arctan\left(\frac{\mathbf{r}^i(3) - \mathbf{s}^j(3)}{\sqrt{(\mathbf{r}^i(1) - \mathbf{s}^j(1))^2 + (\mathbf{r}^i(2) - \mathbf{s}^j(2))^2}}\right) \\ \arctan\left(\frac{\mathbf{r}^i(2) - \mathbf{s}^j(2)}{\mathbf{r}^i(1) - \mathbf{s}^j(1)}\right) \end{bmatrix} \tag{3.36}$$

For obtaining the measurements, two sensors orbiting Earth are considered with their respective orbital elements shown in Table 3.2. Each sensor has a field of view (FOV)

	Sensor 1	Sensor 2
a (km)	6678	6778
e	0.01	0.002
i (deg)	2	70
Ω (deg)	38	20
ω (deg)	40	30
θ (deg)	111	5

Table 3.2: Sensors' Orbital Elements

half-angle of 5 degrees. The measurement noise covariance for each sensor is shown.

$$R_1 = \begin{bmatrix} 0.1^2 & 0 & 0 \\ 0 & (\frac{2\pi}{180})^2 & 0 \\ 0 & 0 & (\frac{2\pi}{180})^2 \end{bmatrix} \begin{bmatrix} km^2 \\ rad^2 \\ rad^2 \end{bmatrix} \tag{3.37}$$

$$R_2 = \begin{bmatrix} 0.05^2 & 0 & 0 \\ 0 & (\frac{1\pi}{180})^2 & 0 \\ 0 & 0 & (\frac{1\pi}{180})^2 \end{bmatrix} \begin{bmatrix} km^2 \\ rad^2 \\ rad^2 \end{bmatrix} \tag{3.38}$$

The plot of the targets and sensors in orbit around the Earth is shown in Figure 3.1. The

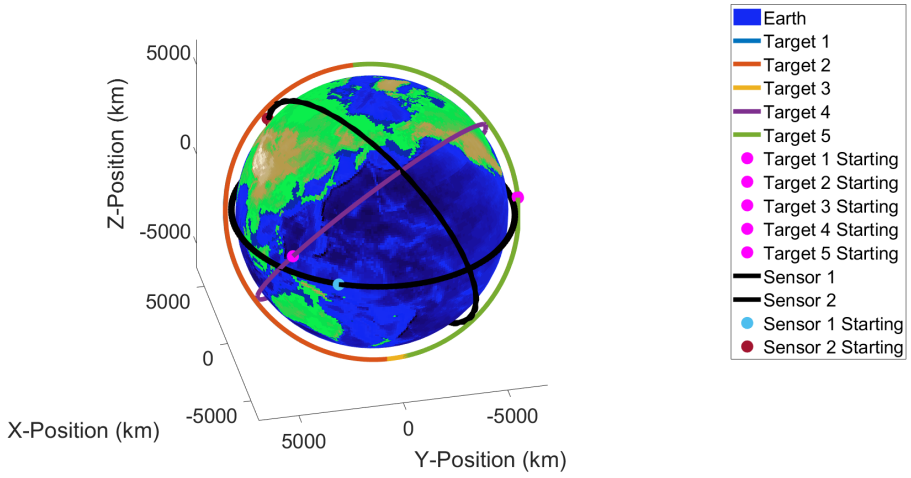


Figure 3.1: Orbits of Sensors and Targets

sensor tasking was done every 5 minutes for a time span of 1000 minutes, however, to incorporate the receding horizon window, groups of 4 time steps were optimized in each iteration. So, instead of 200 iterations, only 50 iterations were done. Some assumptions

that were enforced are: if multiple targets are in the sensor's field of view at a given time instant, the sensor can make a measurement and reduce the uncertainty on all targets, sensors cannot see through the Earth to point at a target, and the distance from a target to a sensor changes the noise covariance matrix, so a large distance will corrupt the quality of the measurement.

3.5.1.2 Resultant Figures

Using the mutual information algorithm, the following sensor schedule was generated for the two sensors, as shown in Figures 3.2 and 3.3. A noticeable pattern in the sensor

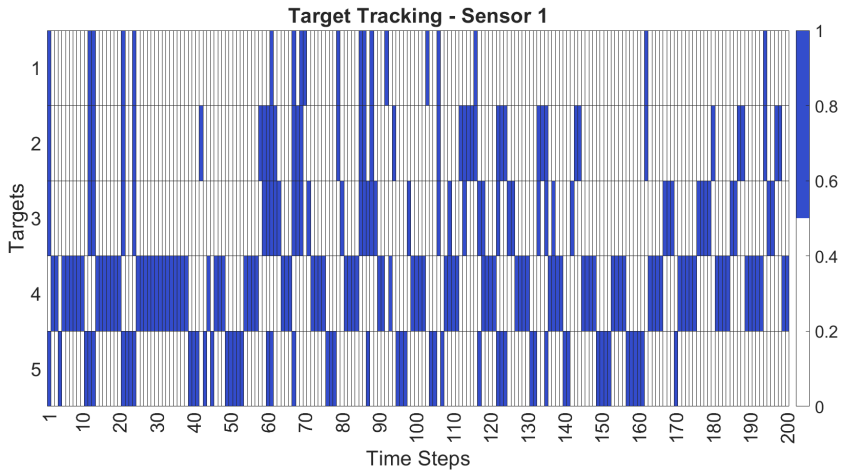


Figure 3.2: Sensor 1 Tasking Schedule for Earth Targets

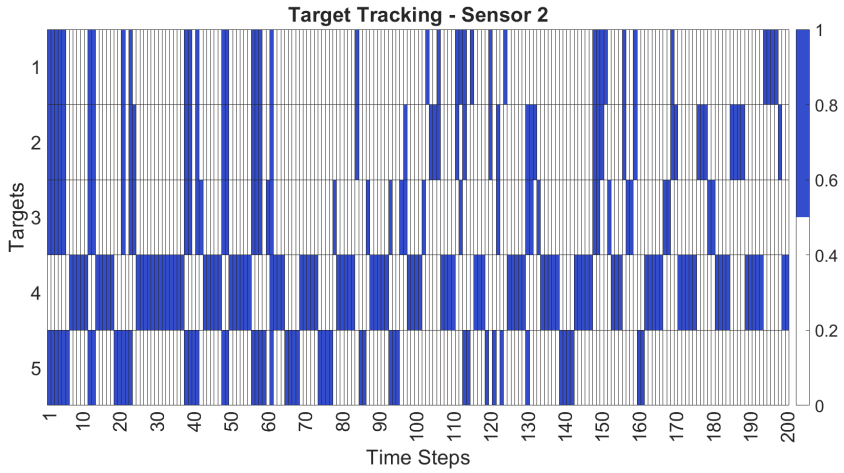


Figure 3.3: Sensor 2 Tasking Schedule for Earth Targets

scheduling is that the sensors observe the same target for several time steps, specifically

in the context of observing target 4. Looking at the initial conditions, target 4 has a significantly different orbit trajectory than the other four targets. Analyzing the geometry of the orbits and the effect of the Earth on obstructing the view on targets, since four of the targets share similar trajectories, the sensors are looking at their only option when observing sensor 4 for quite a few time steps. Applying the sensor schedule, a heatmap of the trace of the position covariance can be generated for each target. Figure 3.4 shows the covariance at each time step assuming no sensor tasking and Figure 3.5 shows the covariance after the sensor schedule shown is applied. It seems that the sensor tasking

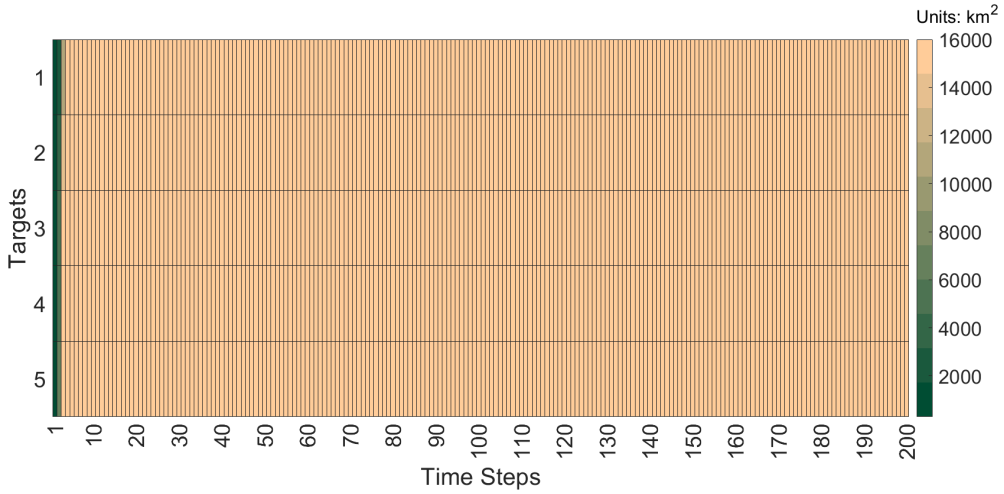


Figure 3.4: Uncertainty Propagation Without Sensor Tasking

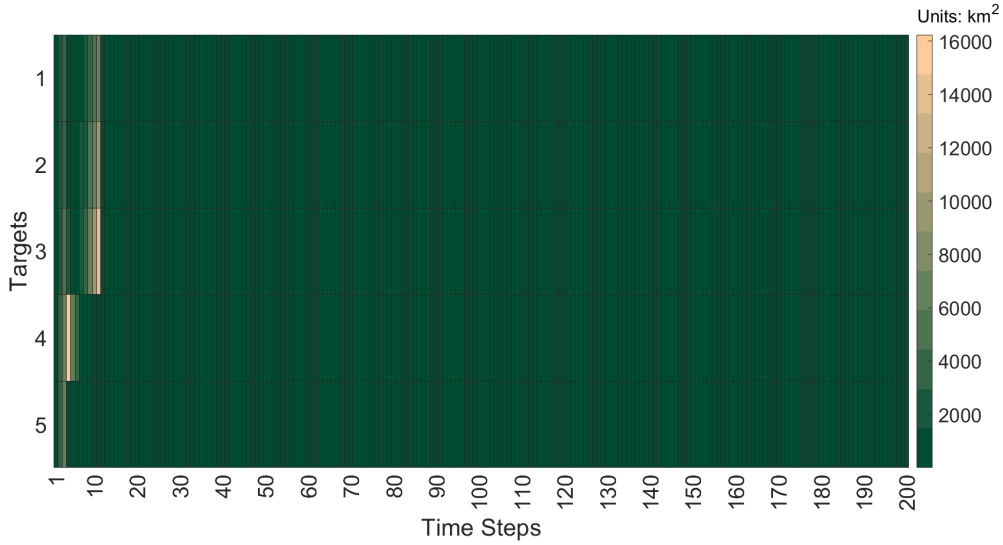


Figure 3.5: Uncertainty Propagation with Sensor Tasking

achieves the goal of reducing the uncertainty of all targets. To confirm that the sensor tasking is working, the error in the mean state vector can be tracked and compared to the bounds of $\pm 3\sigma$, where σ represents the standard deviation. The σ bounds for target 1 are shown in Figures 3.6-3.11 while target 1's true trajectory compared to the sensor tasking trajectory is shown in Figure 3.12.

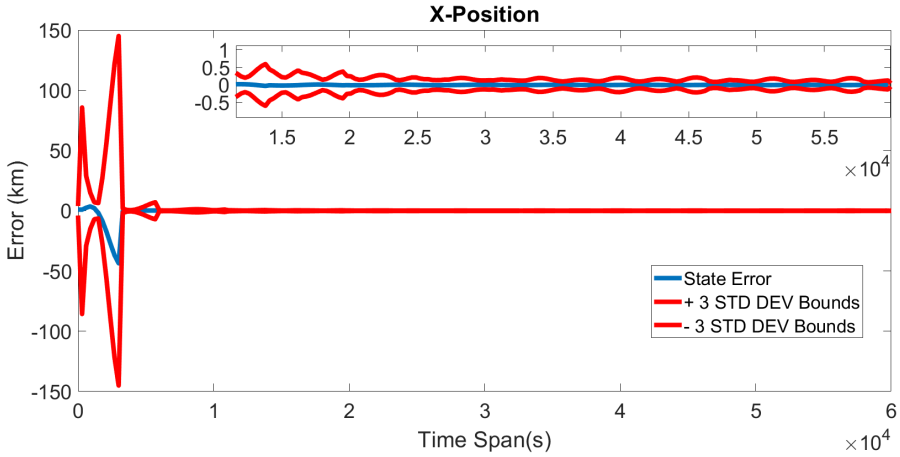


Figure 3.6: Filter Bounds for Target 1 X-Position

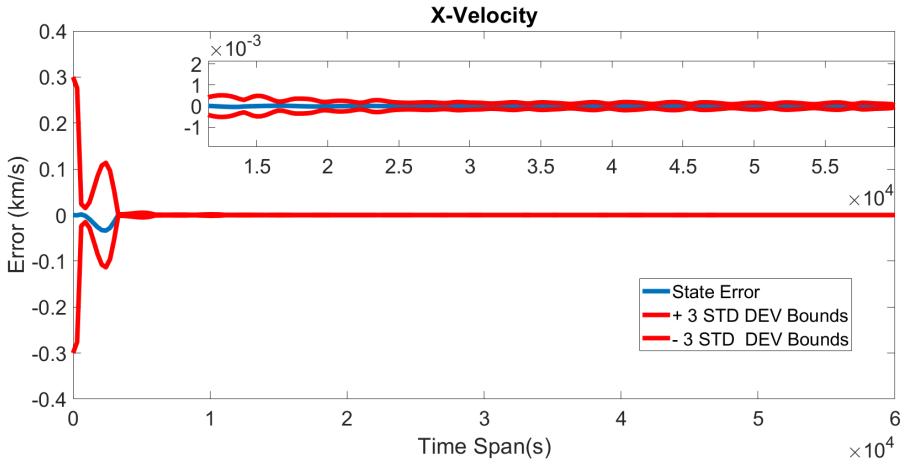


Figure 3.7: Filter Bounds for Target 1 X-Velocity

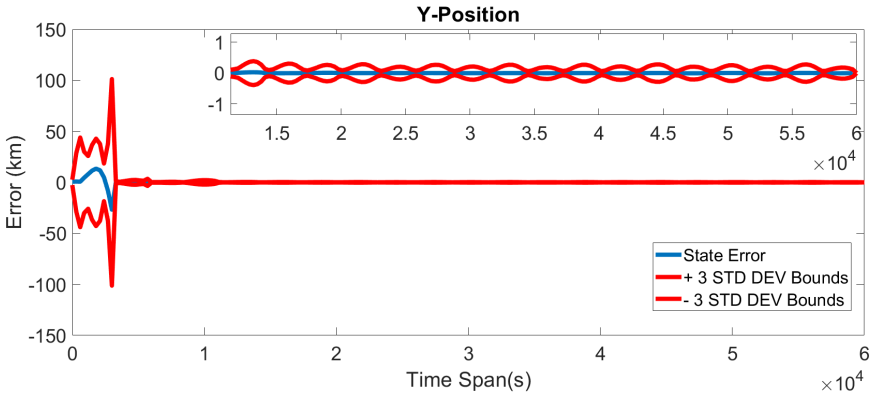


Figure 3.8: Filter Bounds for Target 1 Y-Position

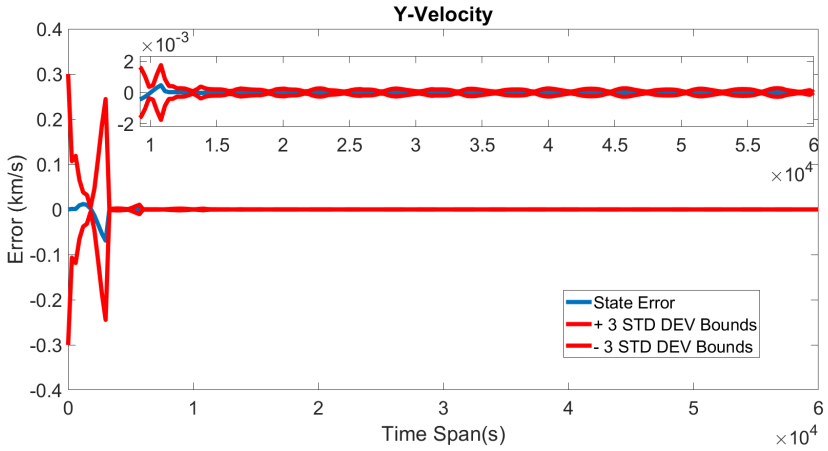


Figure 3.9: Filter Bounds for Target 1 Y-Velocity

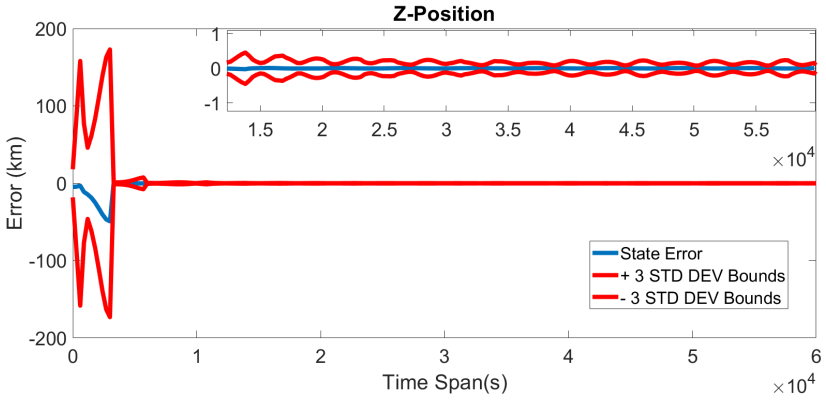


Figure 3.10: Filter Bounds for Target 1 Z-Position

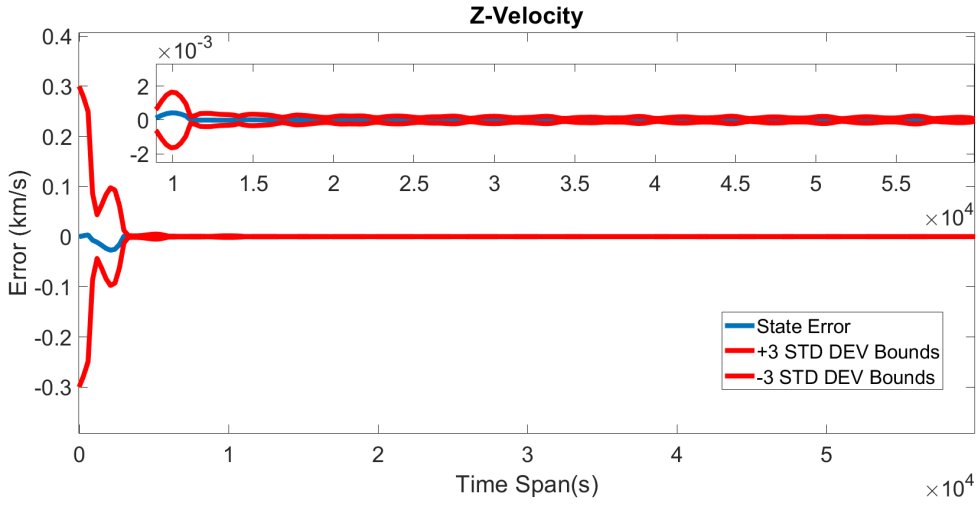


Figure 3.11: Filter Bounds for Target 1 Z-Velocity

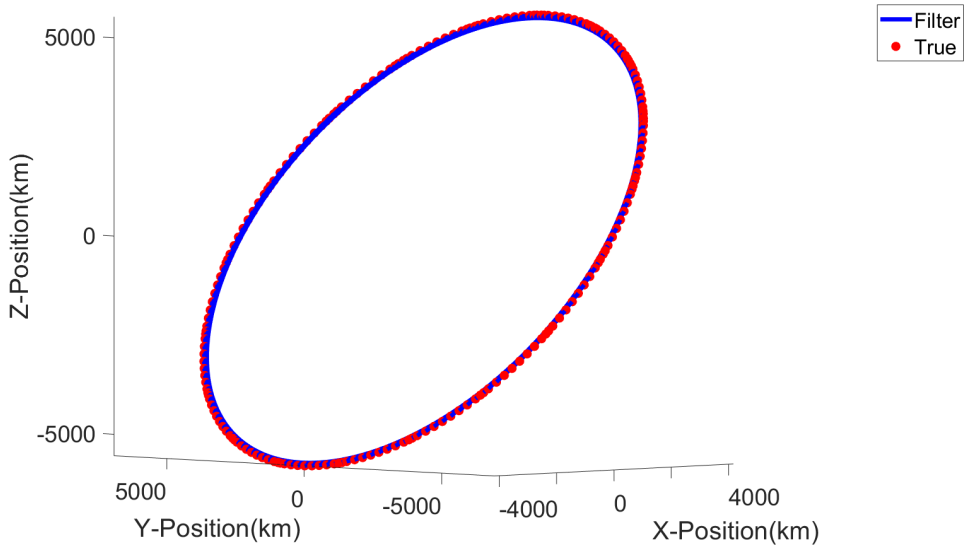


Figure 3.12: True Trajectory vs Filter Trajectory

As can be seen from the figures, the elements of the state vector converge to the mean and the sensor tasking provides an accurate estimate of the true state at each time step. So, the receding horizon window algorithm is functioning properly and can be applied to the cislunar case.

3.5.2 Example 2: Cislunar Target Tracking

Cislunar space involves more complex dynamics than the two-body dynamics employed earlier, specifically the CR3BP dynamics. The CR3BP considers three bodies, where one body is negligible in mass compared to the other two, often called primaries, and the two primaries are restricted to circular orbits around their respective barycenter. More details on the setup of the CR3BP framework can be found in [14] and the example from that paper will be the same one explored here using a receding horizon window approach. The equations of motion that can be found in [14] for the CR3BP dynamics are shown below

$$\ddot{\mathbf{q}} + 2\boldsymbol{\omega} \times \dot{\mathbf{q}} = \frac{\partial \Omega}{\partial \mathbf{q}} \tag{3.39}$$

where $q = [x, y, z]^T$ and $\boldsymbol{\omega} = [0, 0, 1]^T$. $\Omega = \frac{1}{2}(x^2 + y^2) + \frac{1-\rho}{r_1} + \frac{\rho}{r_2}$ and represents the pseudo-potential equivalent for the CR3BP, where $\rho = \frac{m_2}{m_1+m_2}$ is a normalized mass parameter, $r_1 = \sqrt{(x+\rho)^2 + y^2 + z^2}$, and $r_2 = \sqrt{(x-1+\rho)^2 + y^2 + z^2}$. Both r_1 and r_2 represent the distance from the third body to each respective primary. As is convention to nondimensionalize CR3BP quantities [30,31], the characteristic distance is the distance between the two primaries, the characteristic mass is the sum of the masses of the two primaries, and the characteristic time is the time that yields the universal gravitational constant to be unity when performing dimensional analysis.

3.5.2.1 Initial Conditions

There were 21 targets and 2 sensors considered for the cislunar example. The 21 targets have corresponding Jacobi constants and time periods given in Table 3.3. The properties for each target can be used to find the initial condition of each orbit. Each of the targets was propagated for a random duration of their respective orbits using the initial condition to obtain the initial state vector, which consists of position and velocity. The state covariance matrix was also randomly chosen to be one of two possible choices as shown below,

$$P_1 = \begin{bmatrix} 500^2 I_3 & 0_3 \\ 0_3 & 0.01^2 I_3 \end{bmatrix}, P_2 = \begin{bmatrix} 1000^2 I_3 & 0_3 \\ 0_3 & 0.1^2 I_3 \end{bmatrix} \tag{3.40}$$

where the units for the diagonal elements of the covariance matrices are $[m^2, m^2, m^2, \frac{m^2}{s^2}, \frac{m^2}{s^2}, \frac{m^2}{s^2}]$. Process noise was again not included.

Both of the sensors are in-orbit in cislunar space with sensor 1 being in an L_1 Halo Orbit and sensor 2 being in a distant retrograde orbit (DRO). The initial conditions for

Jacobi Constant	Period (TU)
3.1514	3.4141
3.1492	3.41
3.1455	3.403
3.1406	3.3931
3.1344	3.3801
3.127	3.3638
3.1186	3.3438
3.1092	3.3193
3.0989	3.2895
3.0878	3.253
3.076	3.2074
3.0636	3.149
3.0505	3.0706
3.037	2.9562
3.0228	2.7522
3.0155	2.459
3.0158	2.2629
3.0194	2.0674
3.0257	1.8777
3.0344	1.6967
3.0661	1.2937

Table 3.3: Cislunar Targets' Orbit Properties

both sensors are shown below.

$$r_{s_1} = \begin{bmatrix} 0.48488929934701924 \\ -1.2010474350815029 \times 10^{-24} \\ 0.86292128804813717 \\ -1.8187186251023278 \times 10^{-12} \\ 0.50751582888526603 \\ 2.0848313835397631 \times 10^{-12} \end{bmatrix}, r_{s_2} = \begin{bmatrix} 0.36386552085967916 \\ -9.8705964471718881 \times 10^{-23} \\ 2.8928226135819368 \times 10^{-23} \\ 1.2142141762044615 \times 10^{-12} \\ 1.7004305900973176 \\ -2.7836838937168505 \times 10^{-24} \end{bmatrix} \quad (3.41)$$

The initial conditions were generated using [32] by using an orbital period of 13.5639 days for the L_1 Halo orbit and a Jacobi constant of 1.541 and orbital period of 27.38 days for the DRO. Each sensor has a FOV half-angle of 5 degrees. The measurement model consists of range, elevation, and azimuth and is calculated in the same manner as

Equation (3.36). The measurement noise covariance for each sensor is shown.

$$R_1 = \begin{bmatrix} (10^{-6})^2 & 0 & 0 \\ 0 & (\frac{2\pi}{180})^2 & 0 \\ 0 & 0 & (\frac{2\pi}{180})^2 \end{bmatrix} \begin{bmatrix} DU^2 \\ rad^2 \\ rad^2 \end{bmatrix} \tag{3.42}$$

$$R_2 = \begin{bmatrix} (5 \times 10^{-7})^2 & 0 & 0 \\ 0 & (\frac{1\pi}{180})^2 & 0 \\ 0 & 0 & (\frac{1\pi}{180})^2 \end{bmatrix} \begin{bmatrix} DU^2 \\ rad^2 \\ rad^2 \end{bmatrix} \tag{3.43}$$

Figure 3.13 shows the targets' 3-D orbits while Figure 3.14 shows the 3-D orbits of the sensors compared to the orbits of the targets. The sensor tasking algorithm was

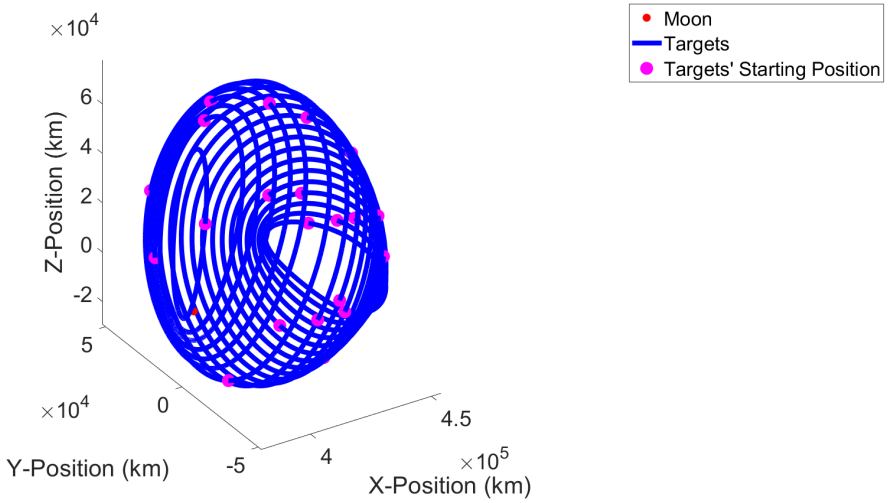


Figure 3.13: Cislunar 3-D Orbits of Targets

implemented for a 15 day period with measurements made every 3 hours. A receding horizon window of 8 time steps was used to task sensors for each 24 hours, so a total of 15 iterations were done. The same assumptions from the previous example were enforced meaning if multiple targets were observed in the FOV of a sensor, the sensor could make a measurement on all of them, sensors cannot look through the moon to make a measurement on a target, and large distances negatively impact the quality of the measurements made by a sensor on a target.

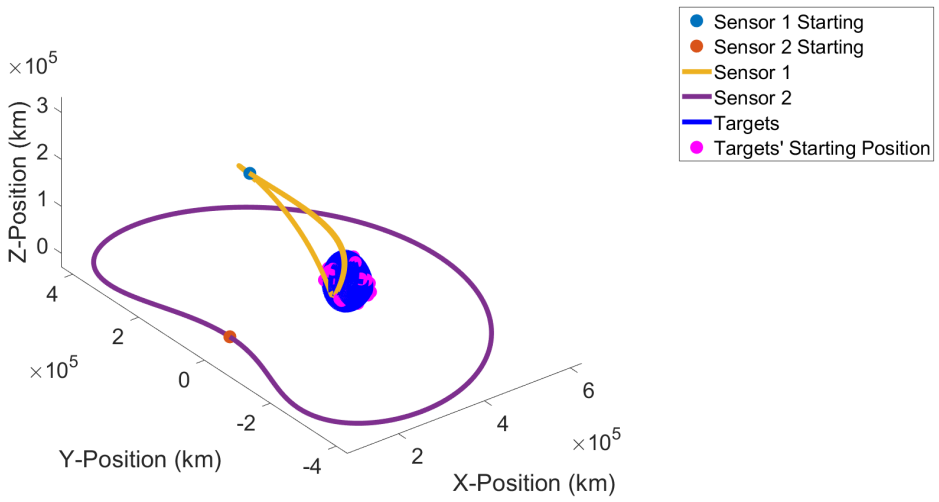


Figure 3.14: Cislunar 2-D Orbits of Targets and Sensors

3.5.2.2 Resultant Figures

Running the mutual information algorithm with the receding horizon window approach, the schedules for both sensors were generated, as shown in Figures 3.15 and 3.16.

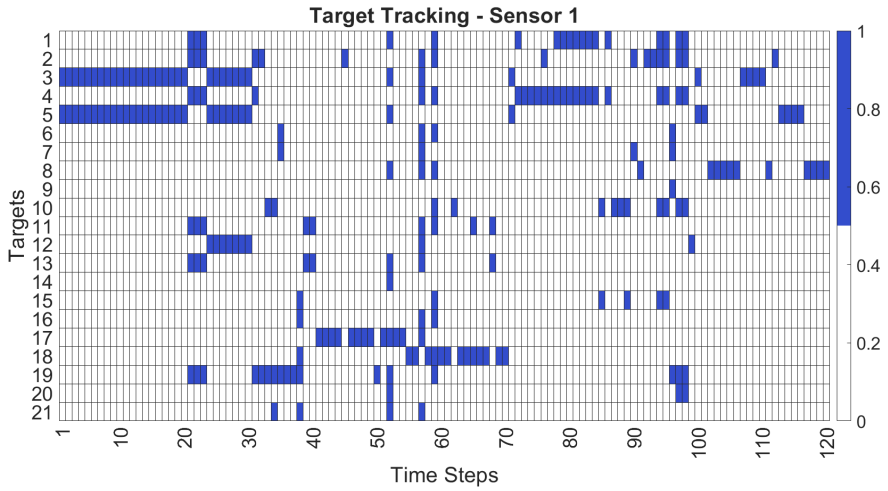


Figure 3.15: Sensor 1 Tasking Schedule for Cislunar Targets

Analyzing the initial orbit trajectories, the middle targets, such as 6-17, are expected to be tasked more frequently as the uncertainty grows more rapidly than the other trajectories. As can be seen from the scheduling, this seems to be the case, especially for sensor 2, as the scheduling heavily prioritizes the middle targets for the first half of the time span. So theoretically, the tasking algorithm is functioning properly. In another

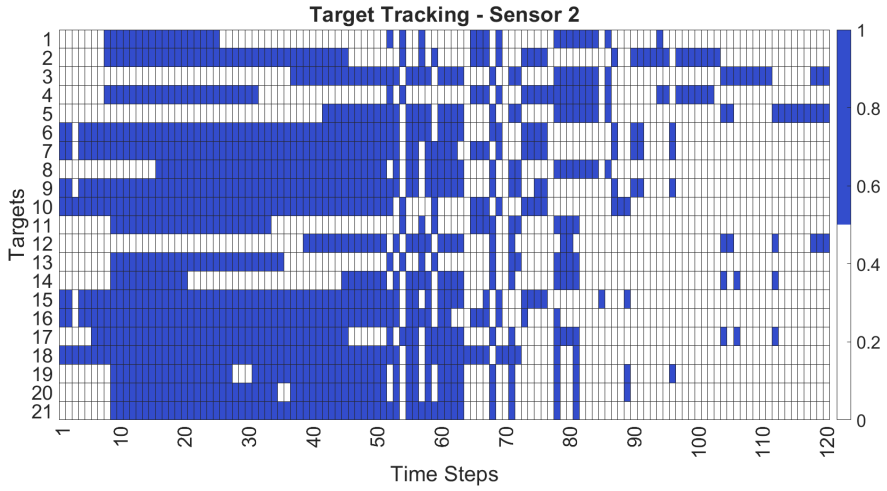


Figure 3.16: Sensor 2 Tasking Schedule for Cislunar Targets

regard, sensor 1 seems to track less objects overall. This could be due to the nature of its orbit as the moon and larger distances could cause the sensor to look at certain targets for multiple time steps, whereas sensor 2 could be able to view multiple targets at once since its view is not always obstructed by the moon.

Similar to the Earth example, applying the sensor schedule can generate a heatmap of the trace of the position covariance for each target along the time span. Figure 3.17 shows the uncertainty propagated without the sensor scheduling and Figure 3.18 shows the sensor scheduling generated applied to the uncertainty propagation. A point to note is that the units for the covariance is DU^2 where 1 $DU = 3.844 \times 10^5$ km.

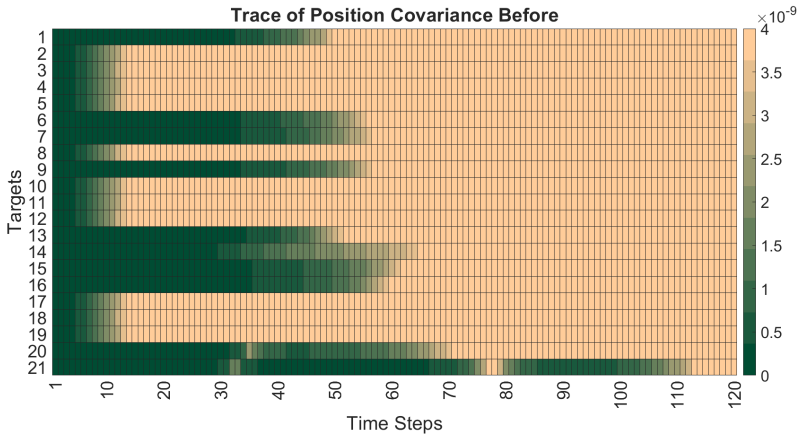


Figure 3.17: Uncertainty Propagation Without Cislunar Sensor Tasking

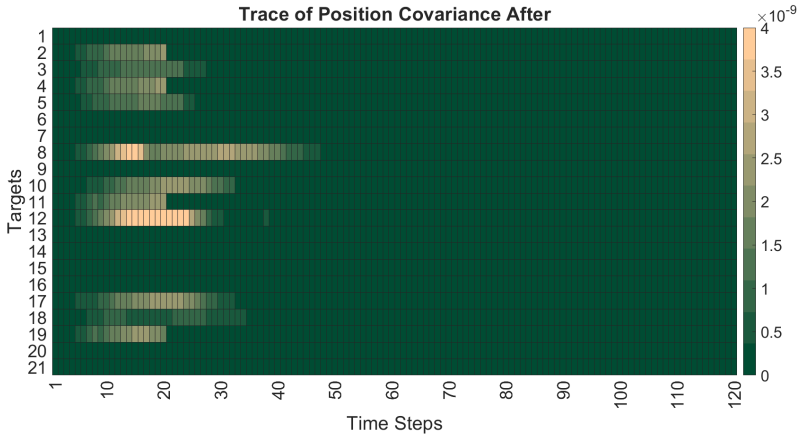


Figure 3.18: Uncertainty Propagation With Cislunar Sensor Tasking

As seen, the position covariance error for all targets does decrease and it seems that the receding horizon window algorithm is working properly. To further guarantee the scheduling is functioning, the filter standard deviation bounds can be plotted again similar to the Earth example as well as the position estimate compared to the true position. For the filter bounds, only the velocity plots will be shown as cislunar space experiences large changes in the state from small changes in velocity. Also, the covariance and position graphs will be used as primary analysis for the position uncertainty being reduced. Figures 3.19 through 3.21 show the velocity filter bounds for Target 5, where 1 $TU = 4.348$ days.

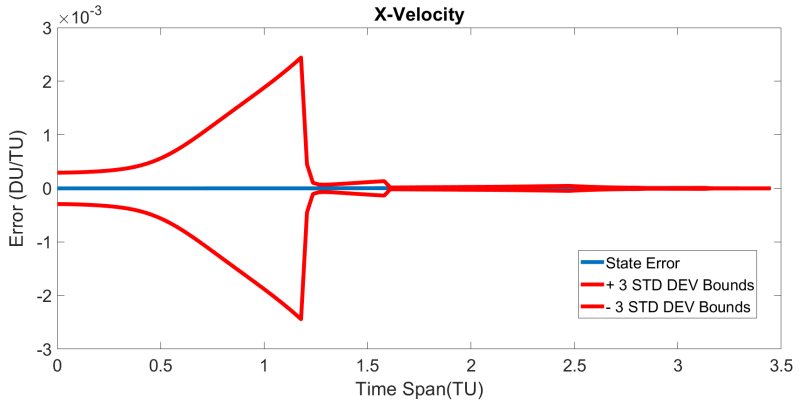


Figure 3.19: Filter Bounds for Target 5 X-Velocity

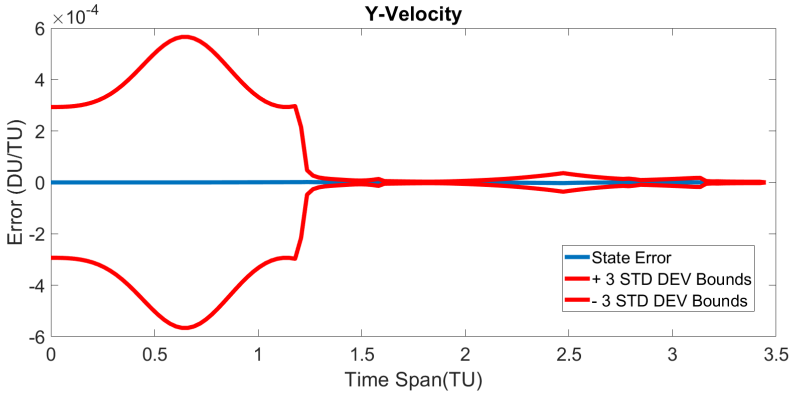


Figure 3.20: Filter Bounds for Target 5 Y-Velocity

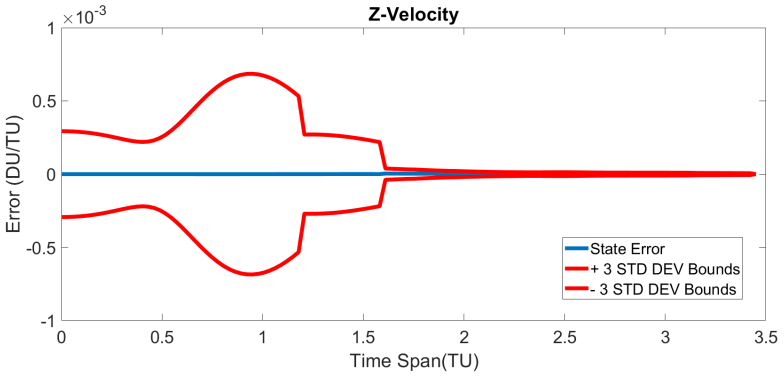


Figure 3.21: Filter Bounds for Target 5 Z-Velocity

Similarly, Figures 3.22 through 3.24 show the velocity filter bounds for Target 21.

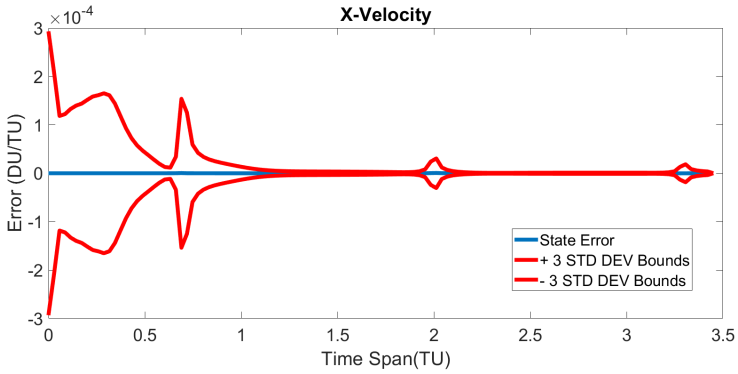


Figure 3.22: Filter Bounds for Target 12 x-Velocity

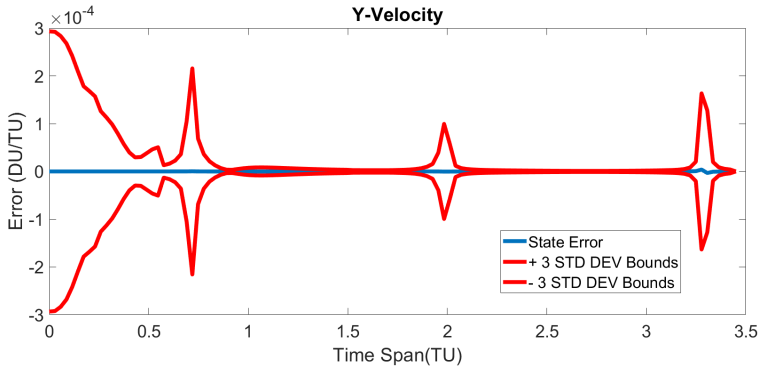


Figure 3.23: Filter Bounds for Target 12 Y-Velocity

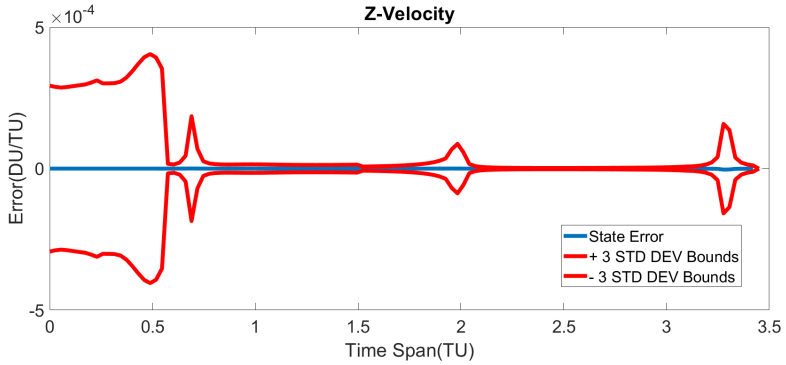


Figure 3.24: Filter Bounds for Target 12 Z-Velocity

Figures 3.25 and 3.26 show the position estimates of each of the two targets.

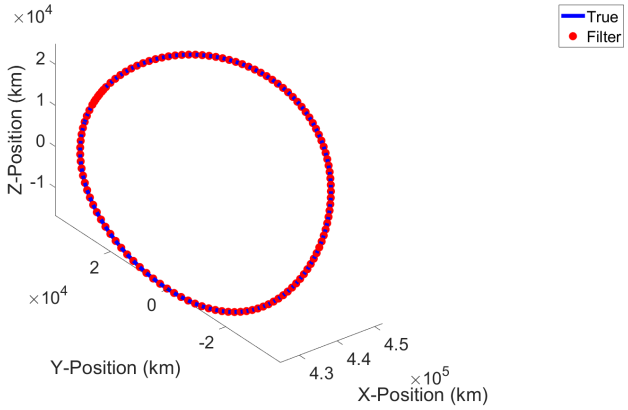


Figure 3.25: Cislunar Target 5 True Trajectory vs Filter Trajectory

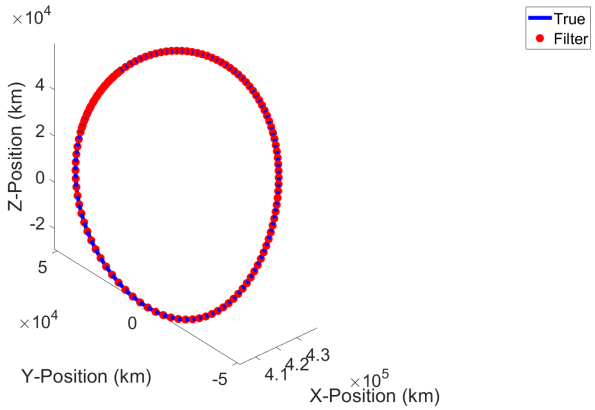


Figure 3.26: Cislunar Target 12 True Trajectory vs Filter Trajectory

As the figures show, the velocity error stays within the filter bounds while the filter bounds decrease for both targets meaning that the errors are acceptable and decreasing. The position graphs further prove this point as the filter and true trajectories are nearly identical meaning that the tasking was able to successfully decrease the uncertainties before the uncertainties became large enough to significantly affect the state estimate.

3.6 Summary

This chapter introduced the concept of mutual information as a sensor performance metric for state estimation. The connection between the CUT filter and mutual information was detailed. A brief explanation of the relation between mutual information and the Kullback-Leibler divergence was given. The mathematical framework for applying a mutual information algorithm for all targets and all sensors over all time steps was developed by deriving the joint mutual information expression. Decision variables were introduced to represent the choice of picking certain sensors at each time step based on which choice maximizes the mutual information expression. The expression was developed so that it could be input into a binary integer programming solver to minimize computation time. Since the exhaustive search is typically computationally expensive and infeasible for large dimensional problems regardless of using a solver, a sub-optimal method was discussed for ease of computational complexity. The sequential-in-time method was introduced as a precursor to the receding horizon window approach, in which the mutual information problem is reduced to being solved at each time step. The receding horizon window approach is discussed by expanding the sequential-in-time method to solving the mutual information problem at a subset of multiple time steps in the entire time span instead of at a single time step. Finally, numerical simulations were shown in two target tracking examples using the receding horizon window approach. Analysis of the results for both the Earth and cislunar example was done by applying the sensor schedule from the mutual information algorithm through a CUT filter. The resultant state estimate was compared to the true trajectory and the error is shown to decrease over the time span such that the state estimate can be considered accurate.

Chapter 4
Conclusion

4.1 Summary

The objective of this thesis was to utilize CUT based uncertainty propagation in conjunction with information theory to task space based sensors to accurately track targets in the cislunar regime. This provides innovation to current implementation capabilities relating to target tracking algorithms. As the number of cislunar missions increases following the current trend, the need to enhance current space situational awareness (SSA) capabilities is crucial so that the growth of the number of objects left by these missions does not become unchecked. Specifically, this thesis implements the receding horizon window approach to a mutual information algorithm. The mutual information algorithm provides a sensor schedule for a given sensor/target problem that reduces the state uncertainty for all targets over time. This is done by applying a Kalman Filtering process over the entire time span to a full exhaustive search of which sensors should observe which targets at each time step. For cislunar space, there is a need to deal with highly nonlinear systems effectively, such as the circular restricted 3-body problem (CR3BP). To tackle this foundational cornerstone problem, the Conjugate Unscented Transform (CUT) of the Kalman Filter was implemented as non-product quadrature scheme does not suffer from the curse of dimensionality.

Since the optimization process is combinatorial in nature as it depends on the number of targets, sensors, and time steps, sub-optimal methods can be utilized to alleviate the computational complexity of the exhaustive search. A common sub-optimal method that is utilized is a sequential-in-time approach where the mutual information algorithm is iteratively solved for each time step. The main innovation shown in this thesis is the implementation of the receding horizon window, which serves as an intermediate between the full exhaustive search and the sequential-in-time approach. The receding horizon

window method optimizes subsets of time steps within the time span at a time to provide faster reduction in uncertainty for all targets.

Numerical simulations demonstrated this method for two target tracking examples: an Earth-based scenario and a scenario in the cislunar regime. The successful implementation of the approach was validated through the statistical analysis of the sensor's schedule effect on the state vector through time.

4.2 Future Work

The approach demonstrated in this thesis enhances current abilities to accurately track several targets whose motion is dictated by extreme nonlinearities. However, further research can further enhance its implementation. One such research avenue could focus on applying more rigorous constraints to the algorithm to represent real-world sensors or incorporating more space environment conditions such as the Sun or other light sources affecting the tracking capabilities of the sensors in-orbit. Similarly, the CR3BP framework was used to generate the trajectories, so future research could involve simulations using higher fidelity dynamical models such as the general R3BP or the ephemeris model.

Another research direction could analyze the accuracy and the convergence of the results to a greater extent. One example of this could be to compare the results obtained from a sequential-in-t ime algorithm to the results for the receding horizon window, specifically analyzing the effect on the convergence time and the overall accuracy. Further analysis could be done on the specific number of time steps in each subset for the receding horizon window approach and the effect on the convergence, specifically focusing on the costs and benefits of increased number of time steps compared to increased computation time.

References

[1] Adurthi, N., P. Singla, and T. Singh (2018) "Conjugate Unscented Transformation: Applications to Estimation and Control," *Journal of Dynamic Systems Measurement and Control-transactions of The Asme*.

[2] Olson, J., S. Butow, E. Felt, and T. Cooley (2022), "State of the Space Industrial Base 2022: Winning the New Space Race for Sustainability, Prosperity and the Planet," .

[3] Jones, A. (2024), "China to launch Lunar Navigation and Communications Test Satellites," .

[4] CSIS Aerospace Security, . (2023), "Counterspace timeline, 1959 - 2022," .

[5] Williams, C. E. (2024), "Artemis," .
URL `https://www.nasa.gov/feature/artemis/`

[6] Congressional Research Service, . (2023), "S.447 - ORBITS Act of 2023," .

[7] Hall, Z. (2021) *A Probabilistic Framework to Locate and Track Maneuvering Satellites*, Ph.D. thesis, The Pennsylvania State University, University Park, PA.

[8] Crassidis, J. L. and J. L. Junkins (2012) *Optimal estimation of Dynamic Systems*, CRC Press.

[9] Chen, G., B. Jia, E. Blasch, K. D. Pham, D. Shen, Z. Wang, and X. Tian (2015), "Space Object Tracking and maneuver detection via interacting multiple model Cubature Kalman filters - IEEE conference publication," .

[10] Paul, S. N. and H. W. Lee *Sensor Tasking for Low Earth Orbit Objects: Leveraging Space Sensor Data for Ground-Based Optical Observations*.

[11] Siew, P. M., D. Jang, T. G. Roberts, and R. Linares (2022), "Space-based sensor tasking using Deep Reinforcement Learning - The Journal of the Astronautical Sciences," .

[12] Thronson, H. (2010), "Human Exploration Beyond LEO by the End of the Decade: Designs for Long-Duration "Gateway" Habitats," .

[13] Raychaudhuri, S. (2008) "Introduction to Monte Carlo simulation," in *2008 Winter Simulation Conference*, pp. 91–100.

[14] Nandan Paul, S., R. Thomas Eapen, and P. Singla (2024) "Sensor Tasking Strategies for Space-Based Observers in the Cislunar Environment," in *AIAA SciTech Forum and Exposition, 2024*, AIAA SciTech Forum and Exposition, 2024, American Institute of Aeronautics and Astronautics Inc, AIAA, publisher Copyright: © 2024 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.; AIAA SciTech Forum and Exposition, 2024 ; Conference date: 08-01-2024 Through 12-01-2024.

[15] Jazwinski, A. H. (1970) *Stochastic Processes and Filtering Theory*, Academic Press, New York.

[16] Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems," .

[17] Anderson, B. D. O., J. B. Moore, and M. Eslami (1979) "Optimal Filtering," .

[18] Julier, S., J. Uhlmann, and H. Durrant-Whyte (2000) "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, **45**(3), pp. 477–482.

[19] Kullback, S. (1959) *Information Theory and Statistics*, Wiley.

[20] Adurthi, N., P. Singla, and M. Majji (2020) "Mutual Information Based Sensor Tasking with Applications to Space Situational Awareness," *Journal of Guidance, Control, and Dynamics*, **43**(4), pp. 767–789.

[21] Wan, E. A. and R. van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," .

[22] Adurthi, N., P. Singla, and T. Singh (2012), "The Conjugate Unscented Transform-An Approach to Evaluate Multi-Dimensional Expectation Integrals," .

[23] ——— (2013), "Conjugate Unscented Transform Rules for Uniform Probability density functions," .

[24] ——— (2012), "Conjugate Unscented Transform and its Application to Filtering and Stochastic Integral Calculation," .

[25] ——— (2015), "A Conjugate Unscented Transformation Based Approach for Accurate Conjunction Analysis," .

[26] Cover, T. M. and J. A. Thomas (2006) *Elements of Information Theory*, Wiley Series in Telecommunications and Signal Processing, Wiley-Interscience.

[27] Zhang, Y., W. Liu, Z. Chen, J. Wang, and K. Li (2015), "On the Properties of Kullback-Leibler Divergence Between Multivariate Gaussian Distributions," .

[28] Shlens, J. (2014), "Notes on Kullback-Leibler divergence and likelihood," .

[29] Gour, G. and M. Tomamichel (2020), "Entropy and relative entropy from information-theoretic principles," .

[30] Szebehely, V. (1967) "Theory of Orbits," Yale University.

[31] Pavlak, T. A. (2013), "Trajectory design and orbit maintenance strategies in multi-body dynamical regimes," .

[32] Park, R. "Three-Body Periodic Orbits," NASA.
URL https://ssd.jpl.nasa.gov/tools/periodic_orbits.html