NON-LINEAR OPTIMIZATION APPLIED TO ANGLE-OF-ARRIVAL SATELLITE-BASED GEOLOCATION

THESIS

Stephen D. Hartzell, Civilian Student, USAF

AFIT-ENG-T-14-J-7

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-T-14-J-7

NON-LINEAR OPTIMIZATION APPLIED TO ANGLE-OF-ARRIVAL SATELLITE-BASED GEOLOCATION

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Stephen D. Hartzell, B.S.E.E.

Civilian Student, USAF

June 2014

AFIT-ENG-T-14-J-7

NON-LINEAR OPTIMIZATION APPLIED TO ANGLE-OF-ARRIVAL SATELLITE-BASED GEOLOCATION

Stephen D. Hartzell, B.S.E.E.
Civilian Student, USAF

Approved:

//signed//	8 May 2014
Andrew J. Terzuoli, PhD (Chairman)	Date
//signed//	8 May 2014
Maj Marshall E. Haker, PhD (Member)	Date
//signed//	8 May 2014
Richard K. Martin, PhD (Member)	Date
//signed//	8 May 2014
Clark N. Taylor, PhD (Member)	Date

AFIT-ENG-T-14-J-7

Abstract

Geolocation is a common application for satellite systems. This involves estimating an object's location (herein called the subject) based on noisy satellite data. Many geolocation methods exist; however, none are tailored specifically for the unique problems faced by satellite systems. Some satellites are so far from the subject being localized that by the time the satellite receives a signal from the subject it might have moved appreciably. Furthermore, some satellites or terrestial sensors may be much closer to the subject than others. Therefore, sensors may need to be weighted based upon their distance to the subject being localized. In addition, even if a subject can be localized, the confidence in this localization may be unknown. Non-linear optimization is proposed, implemented, and analyzed as a means of geolocating objects and providing confidence estimates from passive satellite line-of-sight data. Non-linear optimization requires an initial estimate. This estimate is provided by a triangulation method. The non-linear optimization then improves upon this estimate iteratively by finding estimates that are more likely to have produced the observed line-of-sight measurements. The covariance matrix of the geolocation parameters being estimated is naturally produced by the optimization which provides quantified confidence in the geolocation estimate. Simulations are developed to provide a means of evaluating the performance of the non-linear optimization algorithm. It was found that non-linear optimization can reduce the average error in geolocation estimates, provide improved estimation confidence, and accurately estimate its geolocation confidence for some subjects. The results from the theoretical development of the non-linear optimization algorithm and its simulated performance is quantified and discussed.

To my wife, children, and parents.

Acknowledgments

I thank my wife and best friend first and foremost. Her patiance, encouragement, and love kept me devoted to my studies so that I could help build the foundations of a sound future. Thanks to Dr. Terzuoli for being willing to take on a young and unproven student working on his undergrad. The opportunity provided by Dr. Terzuoli has provided a life that otherwise would have been unlikely. Another special thanks to Dr. Taylor for all the extra one-on-one time spent helping me learn and pointing me to resources to help me develop this research. Likewise thanks to Dr. Martin for his fantastic classes. His expert use of class time and teaching style made the material stick. He made sure that we understood why the conclusions in his classes were true. He made difficult material intuitively make sense. Further thanks to Dr. Haker. His enthusiasm for my research helped keep me motivated. He was also kind enough to offer an independment study during a busy quarter tailored to assist me in learning valuable concepts for my research and career. Lastly, I give thanks to professor Wilson of Sinclair Community College for driving me to be rigorous in my mathematical studies, question everything, and to know why each item covered in my classes is true. He taught and drove me to understand mathematics and science and know why its conclusions were true rather than memorizing its conclusions. I also thank Lee Burchett for helping this research get started and for assisting me when I had theoretical questions. Thanks to Matthew Ryan for helping reduce the the size of the polynomial used in the final non-linear optimization algorithm. Thanks to Jonathan Mautz and associates for assisting in the development of this research project. This project was supported in part by an appointment to the Research Participation Program at the Air Force Institute of Technology, administered by the Oak Ridge Institute for Science and Education.

Stephen D. Hartzell

Table of Contents

Page

List of Figures

Figure Page

List of Symbols

Symbol	Definition
α	angular argument to $\mathbf{e}(\cdot)$ representing the position at the end of the space vehicle's reentry phase
ϵ	Normalized (State) Estimate Error Squared
θ	angle in the xy plane
λ	an eigenvalue
ξ	expected error in a geolocation estimate
ρ	expected precision in a geolocation estimate
μ	mean of quantity in subscript
$\mathbf{\Sigma}$	covarince matrix ($2N \times 2N$)
ϕ	angle coming off the positive z-axis
ψ	Cartesian unit vector representing line-of-sight
ω	angular velocity of $\mathbf{e}(\cdot)$
$\mathbf{\Omega}$	matrix of line-of-sight measurements ($N \times 2$)
c	speed of light
d	shortest distance from line-of-sight to emitter location
D	sum of weighed squares of d
$\mathbf{J}$	Jacobian matrix ($N \times 3$) or ($N \times 6$) for position and velocity
$\mathbf{L}$	lower triangular matrix
M	number of samples in MCS
N	number of line-of-sight measurements
$\mathbf{s}$	emitter state vector 3×1 containing a sensor position
t	time variable
w	weight for the triangulation algorithm
$\mathbf{W}$	weighting matrix

Symbol	Definition
$\mathbf{x}$	emitter state vector (3×1) or (6×1) for position and velocity

Subscripts

0	initial
r	reentry phase
o	orbital phase
d	desired or delay when used with a time variable
e	estimate when used with time variable
f	final
m	measured

List of Operators & Functions

Symbol	Definition
$E(\cdot)$	expected value of argument
$\mathbf{e}(\cdot)$	an ellipse representing the emitter's path
$\|\|\cdot\|\|$	the Euclidean norm
$\tilde{(\cdot)}$	a noisy value
$\hat{(\cdot)}$	best estimate
$\mathbf{p}(\cdot)$	position of subject
$\dot{(\cdot)}$	first derivative of argument
$\mathbf{\Omega}(\cdot)$	noiseless line-of-sight associated with argument
t_d	time delay associated with argument

List of Acronyms

Acronym	Definition
AOA	Angle-of-Arrival
DOA	Direction of Arrival
ECEF	Earth-Centered Earth-Fixed
GEO	Geosynchronous Earth Orbit
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
HEO	High Earth Orbit
IID	Independent and Identically Distributed
LEO	Low Earth Orbit
LOS	Line-of-Sight
MAE	Mean Absolute Error
MCS	Monte Carlo Simulations
MEO	Medium Earth Orbit
NEES	Normalize (state) Estimation Error Squared
NLO	Non-Linear Optimization
SV	Space Vehicle
SVD	Singular Value Decomposition
TA	Triangulation Algorithm
TDOA	Time Difference of Arrival

NON-LINEAR OPTIMIZATION APPLIED TO ANGLE-OF-ARRIVAL SATELLITE-BASED GEOLOCATION

I. Introduction

This chapter introduces the motivation behind this research. It describes the problem at hand and the associated research question. It discusses the assumptions and limitations to this research. Lastly, it provides an outline for the rest of the thesis.

In this research, Non-Linear Optimization (NLO) algorithms are developed and investigated as a means of localizing an object from Angle-of-Arrival (AOA) data. *Localizing* an object means to estimate its position in space. For the purposes of this research, the object which is being localized will be referred to as the *subject*. While AOA data from any source may be used, this research focuses on data passively obtained by satellites. Thus, the sensors do not transmit and receive an echo from the subject. Rather, the sensor observes an emission from the subject. Localization provides an estimate of the subject's location, but without an estimate of the confidence in the subject location, this information may have limited utility.

Therefore, NLO is also investigated for its ability to estimate the confidence it provides in its localization. Furthermore, the algorithm must provide a means of intuitively conveying this information to a user via some visualization.

To conduct this research, the NLO's ability to localize an object and represent its error is theoretically developed. The theoretical performance is compared to its performance in simulated scenarios. A *Scenario* consists of a subject, observing sensors, and their measurements. Two types of scenarios are produced: scenarios for verification and performance evaluation.

The first type of simulation is used to verify that the NLO's theoretical performance matches its simulated performance. NLO is based on several assumptions. Therefore, in scenarios where these assumptions are true, the theoretical results and simulated results should match. This match verifies that the NLO has been correctly implemented. Furthermore, these types of simulations are useful for comparing the NLOs. The NLOs include different assumptions which add to their complexity. Thus, these results may illustrate how much each assumption separately impacts their ability to localize the subject and represent the confidence in their geolocation estimates.

The second type of simulation models a scenario where the NLOs' assumptions are imperfect. Therefore, these simulations are used to evaluate the performances of NLOs in scenarios where their assumptions are approximations. The NLOs will produce less accurate estimates if their assumptions are not perfectly true. By comparing the performances of the NLOs in this scenario, the error resulting from unconsidered factors may be investigated.

This research provides the theoretical development for three NLO algorithms of varying complexity for localization using AOA data. It describes how these algorithms are verified and how their performances are evaluated. The performance and development of the confidence visualization scheme are also given. The next section describes how this research started and why it is a relevant effort.

1.1 Background

In many applications, it is important to estimate the positions of objects and provide the accuracy in these estimates via a process commonly referred to as geolocation or localization [23], [27], [34]. Geolocation provides information on objects world-wide. One of the most common means of estimating the position of an object is via the use of a Global Navigation Satellite System (GNSS) such as the Global Positioning System (GPS) [16]. Each GPS satellite transmits a signal that includes the time the signal was sent, along with

the satellite's position at that time. A receiver may then use this information from multiple satellites to produce an estimate of the object's position by a method known as trilateration [26].

There is an inverse problem of sorts known as passive geolocation or source localization [10]. In this problem, the emitter's position is unknown and must be estimated using measurements from multiple receivers. It is often assumed that the emitter does not transmit its location. Such an emitter is said to be non-cooperative. In this research, the subject is a non-cooperative emitter. Because the subject is non-cooperative, the receivers must produce the information needed to localize the source.

Therefore, differences in the signals received by multiple observing sensors must be exploited for source localization. The two kinds of data are used predominantly in passive geolocation: Time Difference of Arrival (TDOA) and AOA (or equivalently Direction of Arrival (DOA)) measurements [4], [11]. TDOA techniques use the differences between the times when an emitter's signal is received by multiple sensors to perform localization [29]. AOA methods use the direction or Line-of-Sight (LOS) to the emitter. If these LOS are extended, they should meet at a point. Therefore, the point nearest to the intersection of the LOS may be used to estimate the emitter's location.

Passive geolocation is useful in many applications. For example, it may be useful where GPS is not available. It also provides a method for geolocating non-cooperative subjects [29], [32]. The ability to localize non-cooperative subjects has led to research in satellite-based passive geolocation systems that use LOS information [17], [33]. LOS geolocation must be used in some cases where the TDOA cannot be determined. For example, if a signal is emitted continuously, then the start of the signal may be unknown. The phase of the signal could also be exploited to determine the TDOA; however, if the frequency of the signal is too high, then TDOA methods may be impossible. Therefore, AOA geolocation techniques may provide geolocation when other methods cannot be used.

There are several standard LOS geolocation schemes. These often employ some form of triangulation [20], [29]. While these schemes are theoretically and computationally simple, they do not account for the unique challenges faced by satellite geolocation systems. These challenges are discussed below.

Challenge 1: the further a sensor is from the subject, the more an error in its LOS produces an error in geolocation. This fact is illustrated in Figure 1.1. Errors in the LOS measured by sensors that are close to the subject produce smaller geolocation errors than sensors that are further from the subject. This issue is particularly relevant for satellite-based sensors where the distances involved may be large. Thus, the closer sensor should be weighted more heavily than a far-distant satellite since its error propagates over a shorter distance. Because the weighting for each sensor depends on its accuracy and the emitter's location (which is unknown), the weights cannot be calculated.

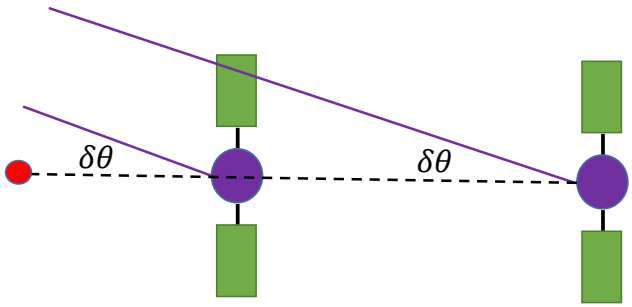

Figure 1.1: Error propagation and triangulation. Both satellites have the same angular error. However, the satellite that is further from the subject (shown in red) has this error propagated over a larger distance.

Challenge 2: Another factor ignored by triangulation is that satellites may be so far from the subject that the subject may have moved a significant distance by the time its emission is detected by the satellites. This problem is referred to as the time delay, and it

is illustrated in Figure 1.2. Therefore, the satellite's LOS is a measure of where the emitter used to be rather than where it is when the sensor produced its measurement.

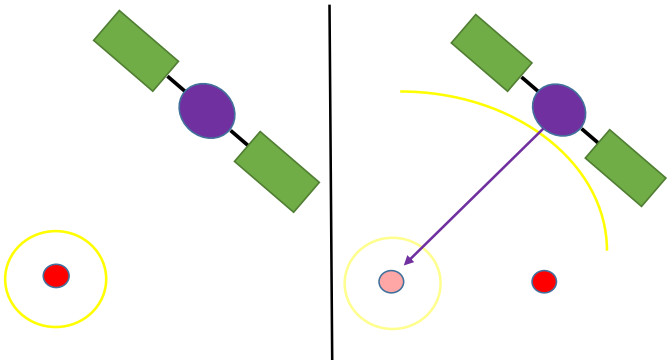

Figure 1.2: The time delay problem. The left-hand frame shows the subjecft when it emitted a signal (shown in yellow). By the time this signal is received by the sensor (shown in frame 2), the subject has moved a significant distance.

Challenge 3: Furthermore, satellites may not produce LOS measurements simultaneously. This problem is illustrated in Figure 1.3. Finding the point nearest the intersection of these LOS assumes that the subject is in one place when these measurement are produced. Each satellite's LOS is measured at different times. For a moving subject, these LOS measurements point toward the subject at different points along its path. Therefore, the subject must be localized at a chosen time. Thus measurements produced closer to this time should be given more weight.

Challenge 4: Furthermore, triangulation may not calculate the confidence in its geolocation. Therefore, there is no perfect measure of how accurate or precise the localization may be. For these reasons, it is desirable to find another geolocation method which resolves the issues associated with satellites.

It has been noted [24] that geolocation typically has a nonlinear nature. Thus, the NLO has been considered for geolocation [9]. However, NLO has never been applied to

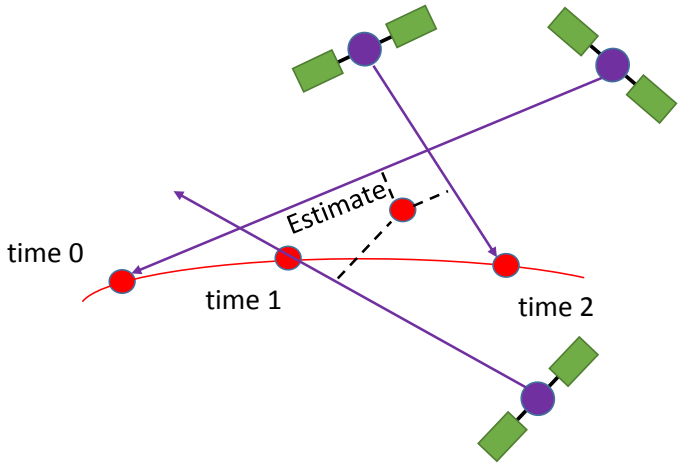

Figure 1.3: Triangulation problem of asynchronous LOS measurement times. The LOS to the subject has been measured at three separate points along its track (the red line).

geolocating satellite systems using LOS data. In addition, NLO's ability to resolve the aforementioned shortcomings has not been investigated.

1.1.1 Research Question & Significance.

The focus of this research is to investigate two queries: First, how well does NLO geolocate the subject in a satellite system that measures AOA? Several NLOs might be used (three are considered here). This question then extends to asking which NLO(s) perform the best. Secondly, is there a useful means of estimating and visualizing the confidence in the geolocation estimates it provides? If the NLO is capable of overcoming triangulation's shortcomings, it may be able to provide users with several benefits.

First, if NLO compensates for the sources of error in satellite geolocation, then position estimates will be more accurate. The NLO may also provide better confidence than triangulation in its estimate, thereby providing a benefit to the user. This improved

confidence may allow for other, more accurate, sensors to hone in on the emitter more quickly.

1.2 Research Assumptions and Domain

This research involves multiple assumptions and practical limitations. In addition, this research evaluates the research question only for a specific set of scenarios. The following section discusses the research assumptions.

1.2.1 Assumptions.

For most passive geolocation techniques, knowledge of the sensors' positions (satellites for the purposes of this research) is requisite. In reality, there may be some uncertainty in sensor locations. For the purposes of this research, this issue is ignored and it is assumed that the satellites' positions are known absolutely.

The Lines-of-Sight (LOS) to the subject is also uncertain. Measurements typically have two sorts of errors: systematic and stochastic errors. In this research, it is assumed that the errors are stochastic and uncorrelated. Stochastic errors are typically described by a probability distribution function. A Gaussian distribution is used as a common first assumption. This distribution is assumed for random variables in this research.

The distribution is described in mathematical detail in Chapter 2. It is also assumed that these measurements are Independent and Identically Distributed (IID). That is, each measurement is independent of the measurements which came before and after it. Gaussian distributed random variables are often assumed to be IID.

In addition to assumptions about the measurements and the sensors, assumptions are made about the subject. Geolocation performance may vary with the type of subject. To simplify this research, only one type of subject is considered. Given the discussion on the sources of error discussed in Section 1.1, it is assumed that a Space Vehicle (SV) is the most challenging subject on average. An SV is most challenging in the sense that SVs are assumed to result in worse accuracy and confidence than any other subject on average. SVs

produce the worst geolocation performance because they are both the fastest subjects that might be observed and their flight paths are nonlinear.

It is also assumed that the times at which each satellite receives a signal is exactly known. This assumption is justified for the following reason: the times at which the satellites produce their measurements matter only because the emitter may be in motion. Therefore, if the reception time is off by some amount, then the emitter may have moved some distance during that time. Given the accuracy of satellite clocks, even the fastest emitter could not move an appreciable distance over the duration of the timing error. In addition to these assumptions, there are practical constraints and limitations.

1.2.2 Limitations.

NLO is fundamentally limited in that it optimizes over a model of the scenario. The model describes the mathematical relationship between the subject's motion and satellite positions and the LOS measurements. The model will not perfectly represent the subject's motion.

Real subjects will have velocity, acceleration, jerk, jounce, and other higher order moments. For example, if the NLO is constructed to account for velocity, then it assumes that the emitter is approximately moving in a straight line over the duration of the measurements used for geolocation. Because there is a limit to the number of parameters that can be included, the NLO cannot include arbitrary subject motion in its model. In this research, only the position and velocity of the emitter and the time delay from the emitter to the satellite are optimized. This thesis evaluates whether satisfactory results may be achieved given these NLO's models.

A further practical limitation of this research is real LOS data. No real data was available for this research. Likewise, typical confidence in LOS is unknown. Therefore, the NLO is evaluated on simulated data with an arbitrary confidence in the LOS measurements. The scope is this research is described in the next section.

1.2.3 Scope.

This research is strictly focused on satellite based geolocation. Other platforms could be used in conjunction with a satellite system. The effects of terrestrial sensors on geolocation performance are not considered. Furthermore, TDOA and LOS information could be available. In this case both could be used in conjunction to provide geolocation estimates. This is not considered.

Neither is the full range of possible NLO algorithms. Only three NLO algorithms are considered here. It is anticipated that these three optimize the most significant unique sources of error as previously described.

Another geolocation task that this research ignores is distinguishing LOS information for multiple subjects. It may be ambiguous which LOS measurements relate to which subject, and this difficulty is compounded if the number of emitters is unknown. For example, two emitters could be at nearly the same LOS relative to a satellite, but their distances to the satellite could vary drastically. It would be difficult to determine which emitter corresponds to which LOS.

The scope of this research also excludes the computational speed of the NLO. There may be requirements regarding the speed at which a geolocation algorithm performs. The speed of the NLO is recorded, but it is not used as grounds for evaluating the NLO's performance.

1.3 Evaluation Methodology

To properly evaluate the benefits provided by the NLO, its geolocation performance is evaluated in four ways: the mean absolute error in the geolocation estimate, the bias in its estimate, the confidence in its estimate, and the accuracy of its confidence estimate. These are collectively referred to as *performance* of the geolocation algorithm. These values are anticipated theoretically and calculated in simulated scenarios.

The theoretical properties of the NLO's performance metrics are developed analytically. The theoretical performance is then verified by simulated scenarios in which the NLO's assumptions are exact. Upon verification of the NLOs theoretical performance, the NLO is used in simulated scenarios where the NLO's assumptions are approximately true. The NLO's performances in these scenarios are compared against each other and another common LOS geolocation algorithm: a Triangulation Algorithm (TA). The TA does not take into account the unique sources of error posed by satellite systems. Thus, the comparison between the TA and NLO provides data on the performance increase or decrease provided by optimizing for these sources of error. The comparison between the NLO's performances provides the geolocation improvement resulting from adding model complexity to the NLO.

To estimate the mean absolute error and bias of the algorithms, truth data and noisy LOS measurements are simulated. The truth data are the true LOS to the subject and the subject's position (and velocity if relevant). The TA and NLO use the noisy data, and their geolocation estimates are compared to the truth data. This comparison provides a metric for quantifying the accuracy provided by the NLO and TA.

The confidences provided by the NLOs and TA are described by the variance in their estimates. The comparison between their true confidence and the confidence they estimate is also calculated. This is the accuracy in their confidence estimates. These metrics are calculated from synthesized scenarios. Monte Carlo Simulations (MCS) are used to provide the algorithms' true confidences. The NLOs have a means of calculating their confidence without MCS. Therefore, their confidence estimates are compared against the confidence estimates produced from MCS. The confidence and accuracy of the confidence estimates provided by the NLOs are compared. As mentioned in Section 1.2, three NLOs of various complexity are evaluated. Therefore, the benefit of adding more parameters (and complexity) to the NLO is analyzed.

To find the performance differences provided by the NLOs, the NLO is developed in three stages of optimization. The first NLO optimizes the emitter's position, the second NLO optimizes position and velocity, and the final NLO includes the time delay. This research compares the accuracy and confidence provided by these three NLOs to evaluate the performance difference resulting from the added complexity.

1.4 Overview

Chapter 2 begins with an explanation of the data used by the geolocation algorithms. It proceeds by explaining how simulations create these data and how the most challenging subject, the SV is synthesized. The chapter proceeds by describing and deriving the mathematics behind the TA and general NLO algorithms. The chapter closes by explaining how the confidence estimates provided by the geolocation algorithms are calculated and visualized.

Chapter 3 discusses how the theoretical building blocks described in Chapter 2 come together in simulations. It describes the combined synthesis of the satellites, their LOS measurements, and the subject. Next, it is explained how specific types of scenarios are simulated to verify the functionality of the NLOs. This is followed by an explanation of each NLO and the subtleties of their operation. The chapter closes by discussing how the performance metrics are calculated and why they were chosen.

Chapter 4 provides the accuracy and confidence data produced by the methodology. The methodology is used to verify the functionality of the NLO algorithms and draw comparisons between their performances with each other and the TA. Chapter 5 begins by describing conclusions drawn from the results, discusses the impact of these conclusions, and ends by describing unexplored questions and future work.

II. Background

The mathematical foundation and theory underlying this thesis is presented in this chapter. Section 2.1 defines the coordinate systems used throughout this thesis and describes the theory behind the model for the data. Section 2.2 describes the mechanics behind the flightpath of the Space Vehicle (SV). The Triangulation Algorithm (TA) which is used for an initial estimate of the emitter's location is given in Section 2.3. The linear algebra that the NLO is based on is detailed in Section 2.4. Section 2.5 presents the theory behind the technique for visualizing the confidence in the NLO. Monte Carlo Simulations (MCS) as a means of visualizing the confidence in the TA are given in Section 2.6. This section also explains how MCS are used as truth for geolocation confidence.

2.1 Coordinates and Data

The standard coordinate system that is used throughout this research is Earth-Centered Earth-Fixed (ECEF) coordinates. The definition of ECEF coordinates is described [33] as follows. It is a right-handed coordinate system with the origin at the center of mass of the Earth, the x-axis protrudes at the zero meridian, the z-axis lies along the earth's axis and points toward the North pole, and the y-axis is along the cross product of the z-axis and x-axis. The ECEF coordinate system is illustrated in Figure 2.1.

The LOS data is described by two spherical angles in the ECEF coordinate system. The definition of these angles relative to the ECEF coordinate system is shown in Figure 2.2. In this figure, the magenta line $\boldsymbol{\psi}$ represents an arbitrary Cartesian LOS vector as given by

$$\boldsymbol{\psi} = \begin{bmatrix} \psi_x & \psi_y & \psi_z \end{bmatrix}^\top \tag{2.1}$$

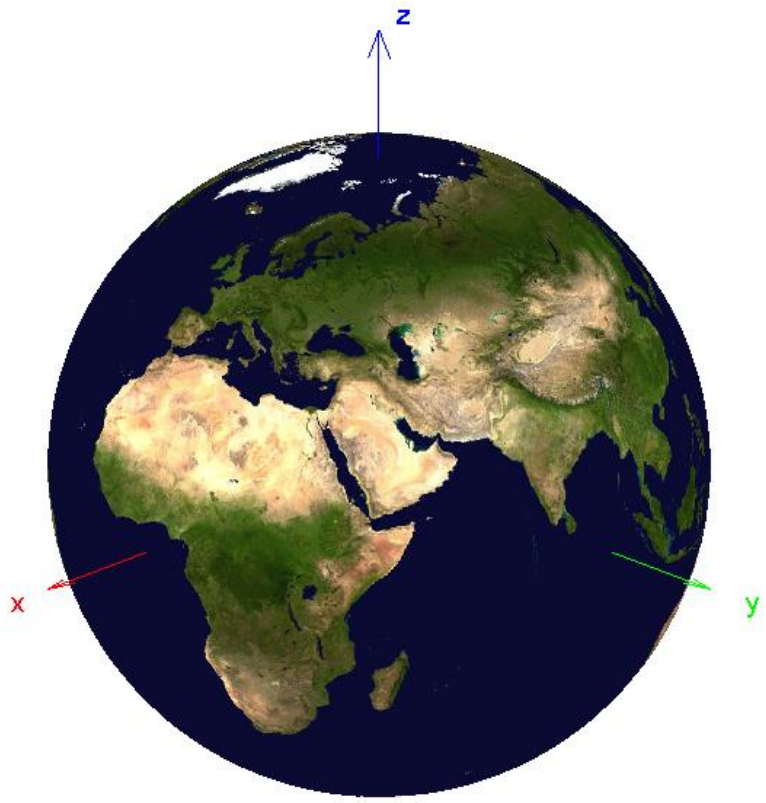

Figure 2.1: Earth-Centered Earth-Fixed coordinates definition

The yellow line represents its projection onto the xy plane. Therefore, θ is the azimuthal angle measured clockwise about the z-axis between the x-axis and the projection of $\boldsymbol{\psi}$ onto the xy plane. ϕ is the zenith angle measured between the z-axis and $\boldsymbol{\psi}$. The conversion from spherical to Cartesian coordinates is given by

$$\begin{aligned} \psi_x &= \sin(\phi)\cos(\theta) \\ \psi_y &= \sin(\phi)\sin(\theta) \\ \psi_z &= \cos(\phi) \end{aligned} \tag{2.2}$$

Note that $||\boldsymbol{\psi}|| = 1$ because LOS measurements describe only the direction to the subject.

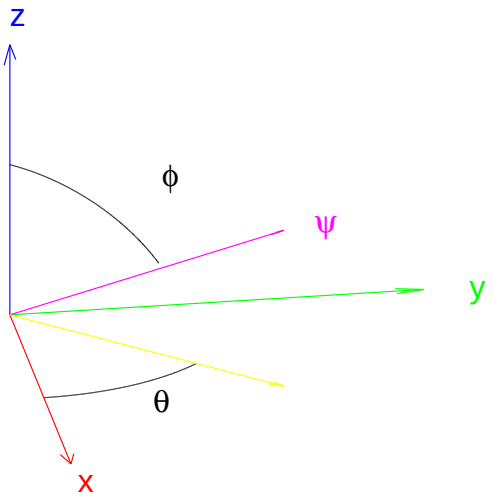

Figure 2.2: Spherical and Cartesian coordinate convention

The conversion from ECEF Cartesian coordinates to spherical coordinates [21] is given by

$$\theta = \tan^{-1}\left(\frac{\psi_y}{\psi_x}\right) \tag{2.3}$$

$$\phi = \tan^{-1}\left(\frac{\sqrt{\psi_x^2 + \psi_y^2}}{\psi_z}\right)$$

It is arbitrarily chosen that AOA or LOS to the subject is measured in spherical coordinates by the satellites. If the positions of the sensor **s** and the subject **x** are given as Cartesian vectors referenced to the origin of the ECEF coordinate system, then the LOS vector is

simply their difference vector. Thus,

$$\boldsymbol{\psi} = \frac{\mathbf{x} - \mathbf{s}}{||\mathbf{x} - \mathbf{s}||} \tag{2.4}$$

There are imperfections in any measurements. Therefore, the AOA measurements are taken to be random. As per the discussion on research assumptions in Section 1.2.1, a Gaussian (or Normal) distribution is chosen for the measurement's distribution. Therefore, the noisy AOA measurements are modeled as

$$\begin{aligned} \tilde{\theta} &\sim \mathcal{N}\left(\mu_\theta, \sigma_\theta^2\right) \\ \tilde{\phi} &\sim \mathcal{N}\left(\mu_\phi, \sigma_\phi^2\right) \end{aligned} \tag{2.5}$$

In these equations μ represents the mean and σ represents the standard deviation of the respective distribution. It is assumed that there is no bias in the AOA. Thus, if the subject were stationary, μ_θ and μ_ϕ are the θ and ϕ angles associated with $\boldsymbol{\psi}$ as given in Equation (2.3). With the coordinate system defined, the flightpath of an SV may be described. The SV described in the following section is used to evaluate the performance of the geolocation algorithms.

2.2 Space Vehicle Path

To use a geolocation algorithm, there must be a subject. There are any number of subjects that might be used. Thus, a particular subject was chosen for simplicity. The Space Vehicle (SV) is used to evaluate the performance of the geolocation algorithms. As discussed in Section 1.2.1, an SV was chosen as the subject because it is expected to be the most challenging subject for geolocation.

An SV could be a dead satellite or space junk. These may be in orbit. For most orbits, acceleration is nearly constant. To make this subject more challenging, a reentry phase was added to introduce rapid deceleration into the SV model. Realistically, an SV would decelerate due to friction in the atmosphere and quickly achieve terminal velocity. To maintain the effects of acceleration for longer periods of time, the SV decelerates until it is at rest at its crash location. Thus, there are two phases to the SV's flightpath: the orbital and reentry phases.

For modeling simplicity, the SV flightpath is modeled in reverse. The SV is modeled in two phases. There is an acceleration phase and an orbital phase. These are modeled separately and put together at the end of this section. The SV flightpath begins with its crash location and then accelerates toward its orbital phase. The orbiting phase is modeled as a section of an elliptical path. The trajectory is simulated using the mathematical development that follows. Figure 2.3 illustrates the SV flightpath geometry. The flightpath is simulated in the xz-plane, and then randomly rotated in 3-D.

The standard form of an ellipse is given by

$$\left(\frac{z - c_z}{a}\right)^2 + \left(\frac{x - c_x}{b}\right)^2 = 1 \tag{2.6}$$

This equation will be used to reach the parametric form of an ellipse which is used throughout the rest of this research.

In Equation (2.6), a and b define lengths of the ellipse's axes, the largest of which defines the length of the semi-major axis. The smaller value is the length of the semi-minor axis. To reach the parametric form, the standard form of an ellipse is parameterized as

$$\begin{aligned} z &= a\sin(\phi) + c_z \\ x &= -b\cos(\phi) + c_x \end{aligned} \tag{2.7}$$

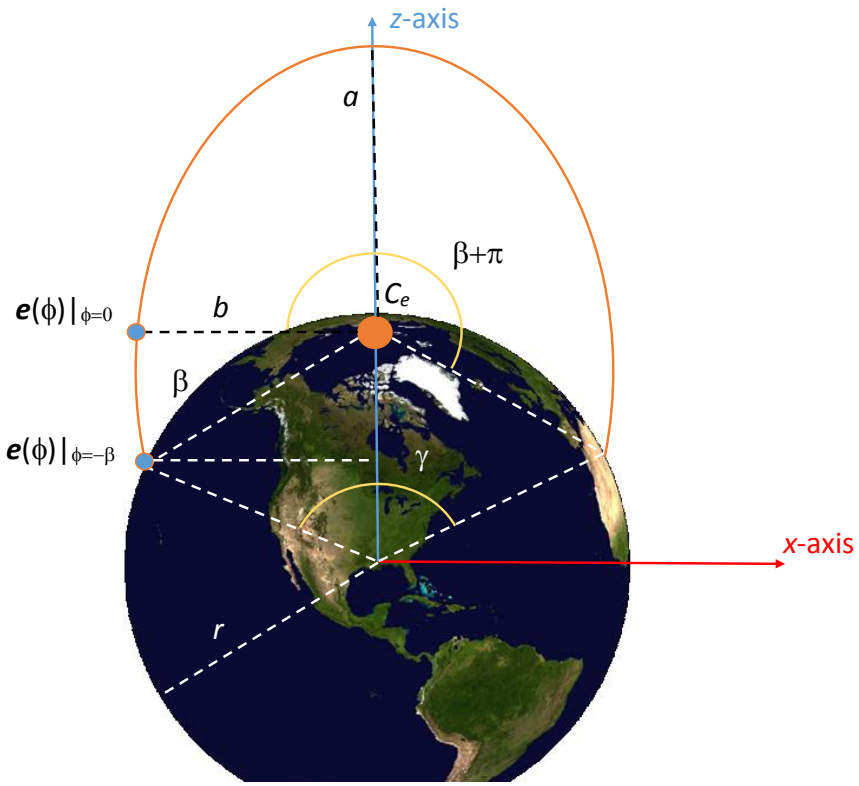

Figure 2.3: Geometry for space vehicle flightpath. Note that for visualization purposes, this figure drastically exaggerates the size of the ellipse.

Substituting the parameterized variables from Equation (2.7) into Equation (2.6) produces

$$\left(\frac{a\sin(\phi)+c_z-c_z}{a}\right)^2+\left(\frac{-b\cos(\phi)+c_x-c_x}{b}\right)^2=1$$

$$\sin^2\phi+\cos^2\phi=1$$

This equation is the trigonometric identity. Therefore, when z and x are parameterized as given in Equation (2.7), Equation (2.6) becomes a tautology. Thus, z and x satisfy Equation (2.6) regardless of the value of ϕ. Therefore, ϕ is a parametric variable for z and

x. An ellipse may then be described as the vector-valued function

$$\mathbf{e}(\phi) = \begin{bmatrix} -b\cos\phi + c_x & a\sin\phi + c_z \end{bmatrix}^\top \tag{2.8}$$

From Figure 2.3, the center of the ellipse C_e lies along the z-axis. Therefore, $c_x = 0$ and Equation (2.8) is simplified to

$$\mathbf{e}(\phi) = \begin{bmatrix} -b\cos\phi & a\sin\phi + C_e \end{bmatrix}^\top \tag{2.9}$$

This elliptical model will be used in Chapter 3 to simulate the SV flightpath. Therefore, the ellipse's parameters must be simulated. The following discussion describes how these parameters are established in the simulations. The following values are randomly generated: C_e, the SV's apogee, and γ. Therefore, one of the ellipse's principle axes may be found via

$$a = \text{apogee} + r - C_e \tag{2.10}$$

where r is the approximate radius of Earth and is given the value 6,378,100 meters.

Next, b is calculated. However, to find b, β must be found first. The z-component of $\mathbf{e}(\phi)|_{-\beta}$ shown in Figure 2.3 is used to find β in terms of other known quantities as given by

$$\begin{aligned} a\sin -\beta + C_e &= r\cos\frac{\gamma}{2} \\ -a\sin\beta &= r\cos\frac{\gamma}{2} - C_e \\ \sin\beta &= \frac{C_e - r\cos\frac{\gamma}{2}}{a} \\ \beta &= \arcsin\frac{C_e - r\cos\frac{\gamma}{2}}{a} \end{aligned} \tag{2.11}$$

β has been determined in terms of known quantities. Therefore, the value of b may be determined by using the x-component of $\mathbf{e}(\phi)|_{-\beta}$ as given by

$$\begin{aligned} -b\cos -\beta &= -r\sin\frac{\gamma}{2} \\ b &= \frac{r\sin\frac{\gamma}{2}}{\cos\beta} \end{aligned} \tag{2.12}$$

Equations (2.10) and (2.12) provide the equations needed to describe the ellipse given in Equation (2.9). Rather than parameterizing the flightpath as a function of ϕ, the flightpath must be described as a function of time t. The SV impact time is arbitrarily defined to be at $t = 0$. (Recall that the SV's flightpath is simulated in reverse.) The point of impact is chosen to be at $\mathbf{e}(\phi)|_{-\beta}$. The flightpath is then modeled as

$$\mathbf{e}(t) = \begin{bmatrix} -b\cos(\omega t - \beta) & a\sin(\omega t - \beta) + C_e \end{bmatrix}^\top \tag{2.13}$$

where ω is the angular velocity of the SV. The flightpath description is nearly complete; however, it does not include the reentry phase. During the reentry phase, the SV starts at rest and accelerates until it reaches its orbiting velocity. Note that because the flightpath is elliptical, the orbiting velocity slightly oscillates. The altitude of a typical SV at the end of its reentry phase is approximately known. Therefore, α is the angle that satisfies

$$a_f = |\mathbf{e}(\alpha)| - r \tag{2.14}$$

This results in a quadratic equation, the solution to which is given in the code given in Appendix A. The value of α will be used for the synthesis of the reentry phase. The reentry phase is given in mathematical form, and will be followed by a discussion of the properties that make this mathematical model attractive.

The flightpath during the reentry phase $\mathbf{e}_r(t)$ is

$$\mathbf{e}_r(t) = \begin{bmatrix} -b\cos\left(\frac{t}{t_f}\omega t - \beta\right) & a\sin\left(\frac{t}{t_f}\omega t - \beta\right) + C_e \end{bmatrix}^\top \tag{2.15}$$

$$\text{where } t_f \triangleq \frac{\alpha+\beta}{\omega}. \tag{2.16}$$

t_f is defined this way because $t = t_f$ when $\omega t - \beta = \alpha$. A realistic orbital velocity v_f is more important for realism than the value of t_f. Therefore, t_f is left as a variable, and a relationship must be found between v_f and t_f. The value of t_f must be found as a function of v_f. The relationship between these values is found by finding the velocity of the SV as a function of time $v_r(t)$, ie

$$\begin{aligned} v_r(t) &= \left|\frac{d}{dt}\mathbf{e}_r(t)\right| \\ &= \sqrt{\left(2b\omega\frac{t}{t_f}\sin(\omega t-\beta)\right)^2 + \left(2a\omega\frac{t}{t_f}\cos(\omega t-\beta)\right)^2} \\ &= 2\omega\frac{t}{t_f}\sqrt{b^2\sin^2(\omega t-\beta) + a^2\cos^2(\omega t-\beta)}. \\ v_f &= v_b(t_f) = 2\omega\frac{t}{t_f}\sqrt{b^2\sin^2\alpha + a^2\cos^2\alpha} \\ \omega &= \frac{v_f}{2\frac{t}{t_f}\sqrt{b^2\sin^2\alpha + a^2\cos^2\alpha}} \end{aligned} \tag{2.17}$$

This equation is used as follows. First, α is found by solving Equation (2.14). Next, Equation (2.16) is used to linearly relate t_f and ω. Finally, ω is found in terms of v_f by Equation (2.17).

The orbital phase is the last portion of the flightpath that needs to be defined. Its definition is

$$\mathbf{e}_o(t) = \begin{bmatrix} -b\cos\left(2\omega t - \omega t_f\right) & a\sin\left(2\omega t - \omega t_f\right) + C_e \end{bmatrix}^\top \tag{2.18}$$

This definition has two key properties. It ensures that the velocity and position of the SV at $t = t_f$ are the same for both phases of the flightpath. This ensures that the transition is

continuous for position and velocity; however, higher order moments such as acceleration and jerk will not be constant. Therefore, the final equation of motion for the SV flightpath is given by

$$\mathbf{e}(t) = \begin{cases} \left[-b\cos\left(\frac{t}{t_f}\omega t - \beta\right) \quad a\sin\left(\frac{t}{t_f}\omega t - \beta\right) + C_e\right]^\top & t \le t_f \\ \left[-b\cos\left(2\omega t - \omega t_f\right) \quad a\sin\left(2\omega t - \omega t_f\right) + C_e\right]^\top & t > t_f \end{cases} \tag{2.19}$$

This final flightpath equation is used to simulate the motion of the subject. The next section describes a typical AOA triangulation method for geolocation.

2.3 Triangulation Algorithm

If extended, LOS measurements from several satellites should nearly intersect. The Triangulation Algorithm (TA) estimates the subject's location as the point where these LOS nearly intersect as illustrated in Figure 2.4. In this research, the TA serves two purposes. First, its performance is compared against the performance of the NLO. This research proposes the NLO as a means of mitigating the factors that the TA doesn't account for such as system geometry, subject motion, and time delay. Therefore, the NLO should provide improved accuracy and confidence in its geolocation estimates than the TA. Second, the NLO requires an initial guess of the subject's position to begin. The TA provides the initial guess.

The mathematical foundation for the TA is similar to that given in [6] and [15]. The following discussion of the TA has been presented in [5]. The first step in this algorithm is to describe the minimum distance d_i to the i^{th} LOS as a function of the emitter's position $\mathbf{x}$. A total distance $D(\mathbf{x})$ is then defined as given by

$$D(\mathbf{x}) = \sum_{i=1}^{N} d_i^2(\mathbf{x}, \boldsymbol{\psi}_i) w_i \tag{2.20}$$

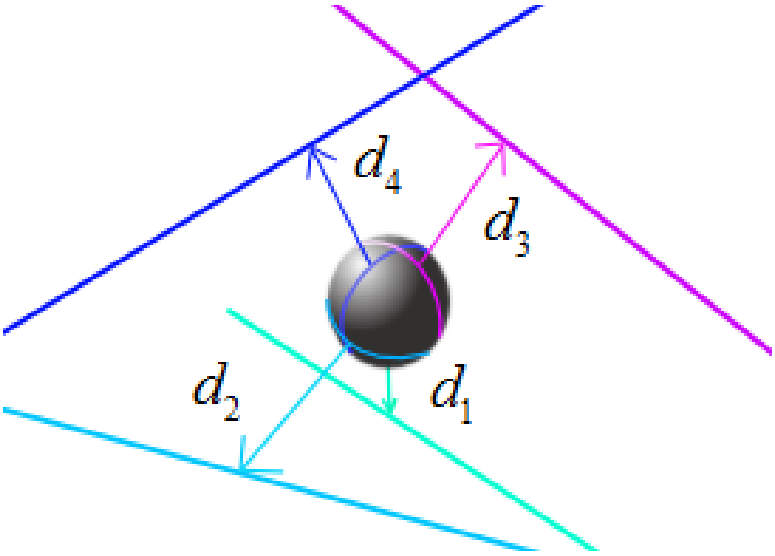

Figure 2.4: Triangulation Algorithm Notion. The Triangulation Algorithm's geolocation estimate minimizes the distances to the lines-of-sight.

where w_i is a weighting term defined on $[0, 1]$ meant to describe the confidence in the i^{th} sensor's LOS measurement.

The TA's geolocation estimate is defined as the point in space that minimizes D. $d_i^2(\mathbf{x}, \boldsymbol{\psi}_i)$ is quadratic in each component of $\mathbf{x}$. D is also positive since it is the sum of squared values. Therefore, D has one minimum. This minimum is the value of $\mathbf{x}$ that is the solution to

$$\nabla D(\mathbf{x}) = \mathbf{0}. \tag{2.21}$$

This equation results in a system of three linear equations in three unknowns. This system may then be solved by any standard linear algebra technique to find the TA's geolocation estimate. The next section describes the mathematical theory underlying the NLO.

2.4 NLO Mechanics

This section introduces the mathematical theory for the NLO. NLO is a means of solving systems of nonlinear equations. It is also known as nonlinear regression in statistics. Specifically, the Gauss-Newton method is utilized in this research though other approaches could conceivably be used. NLO begins with an initial approximation for the solution. The NLO is based on the assumption that a system of nonlinear equations may be approximated by a linear system in the neighborhood of the solution. The linear system is solved for a new estimated solution. It is anticipated that the estimated solution will be closer to the true solution to the nonlinear system of equations than the initial guess.

This process iterates resulting in an estimated solution that converges to the true solution to the nonlinear system. In this way, the analytically intractable problem of solving the system of nonlinear equations is solved to a desired level of accuracy by iteratively solving linear systems of equations.

The following discussion will introduce NLO with a simple case that provides valuable intuition which carries forward into much more complicated versions of the problem. The simple case is a classic nonlinear solver known as Newton's method. Newton's method is developed and discussed in the following section similar to the explanation given in [21].

2.4.1 Newton's Method and Intuition.

Newton's method was originally posed as a means of finding the roots of polynomials of any order. This is useful because there is no analytical means of finding the roots of a fifth or higher order polynomial. Newton's method is applicable to more than polynomials. It can be applied to most differentiable functions. Therefore, for a function $f(\cdot)$, Newton's method may be used to find values of x such that $f(x) = 0$. While Newton's method was originally posed for finding the roots of functions, it is suited to find the values of x that produces any output c from the function. Simply, define some new function $g(x) = f(x) - c$ and use Newton's method to find the roots of $g(\cdot)$.

The first steps in Newton's method are illustrated in Figure 2.5. The objective of Newton's method is to find the input x_d to some function $f(\cdot)$ that produces a desired output $f(x_d)$. Therefore, Newton's method finds the value of x given by

$$\{x|f(x) = f(x_d)\} \tag{2.22}$$

There may be several values of x for which $f(\cdot)$ produces $f(x_d)$. The particular solution that is sought after (the desired value of x) is x_d.

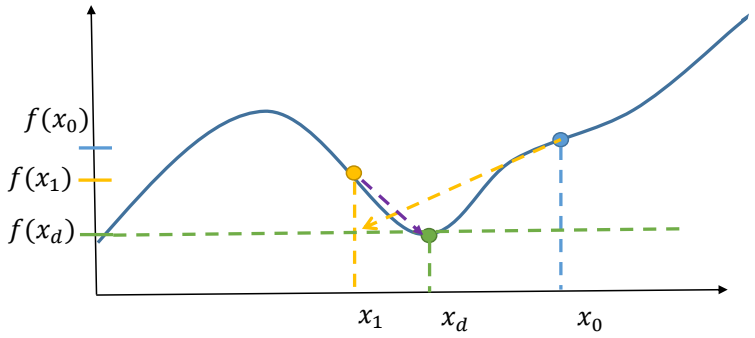

Figure 2.5: Demonstrating the first two steps in Newton's method

In Newton's method, both $f(x_d)$ and an initial guess at x_d called x_0 are known. It is assumed that f is approximately linear in the neighborhood about x_0. Therefore, the function over this region may be approximated by

$$f(x) \approx f'(x_0)(x - x_0) + f(x_0) \tag{2.23}$$

The initial guess is represented by the blue point in the figure. Next, the difference between $f(x)$ and $f(x_d)$, typically referred to as the *residual* Δf, is found. Using the linear approximation and solving this equation for x where $x = x_d$ should produce an estimate of x_d called $\tilde{x}_d$. Equation (2.23) is translated into point-slope form and a more intuitive

notation, ie

$$\Delta f = \frac{df_0}{dx}(\tilde{x}_d - x_0) \tag{2.24}$$

Solving this equation for $\tilde{x}_d$ yields

$$\tilde{x}_d = x_0 + \Delta f \frac{dx}{df_0} \tag{2.25}$$

$f(\tilde{x}_d)$ should be closer to the objective value $f(x_d)$ than $f(x_0)$. $\tilde{x}_d$ is represented in Figure 2.5 by the yellow point. Next let, $x_1 = \tilde{x}_d$. This process now repeats by letting x_1 be the initial guess for the new iteration. This is illustrated by the purple line. The value of $\tilde{x}_d$ will quickly approach the solution Equation (2.22) if locally linear. This iteration stops once the residual Δf has reached some small value chosen by the user.

It should be noted that this technique may fail as a result of using an inaccurate starting point. Two cases are demonstrated in Figure 2.6.

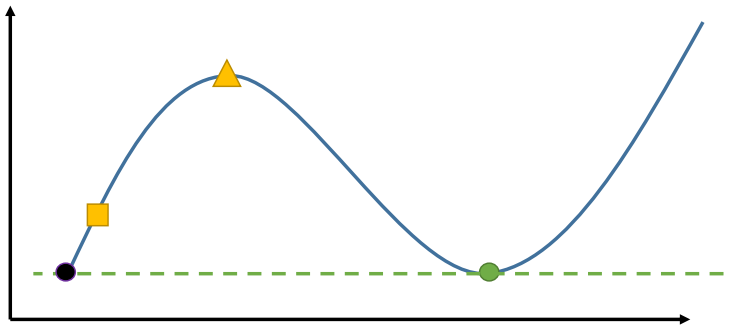

Figure 2.6: Cases where Newton's method fails. The top starting point (the triangle) is at a peak, so the slope is 0 at this point. The lower starting point (the square) will converge to the black point rather than the green point.

In Figure 2.6, the yellow triangle and square are initial guesses, the green point is the desired point, and the black point has the same value of $f(\cdot)$ as the green point. The guess

at the local maximum in this plot is an example of plateauing. Here the slope is almost zero. Thus, the $\frac{dx}{df}$ term in Equation (2.25) is undefined or nearly infinite.

The lower initial guess is another example of a inaccurate starting point. Here Newton's method will converge to the black point rather than the green point. If x_d represents a physical value, then the value of x at the black point may be absurd. For instance, in geolocation, the purple point may represent a subject with a speed greater than the speed of light. Convergence to an undesired solution is the more common cause of failure in Newton's method. For this reason, the initial guess at x_d must be close to the true value.

All of the properties and problems regarding Newton's method have analogous counterparts in NLO. Therefore, the intuition provided by Newton's method carries forward into the NLO problem where systems of nonlinear equations are considered. Like Newton's method, NLO approximates a nonlinear system of equations with a linear system of equations. Linear systems of equations are iteratively solved to approximate the solution to the nonlinear problem.

2.4.2 Linear Least Squares.

NLO operates on an overdetermined system of nonlinear equations. Since a system of nonlinear equations can often not be solved analytically, it is locally approximated by a system of linear equations [30]. In many applications, the system of nonlinear equations is overdetermined. That is, there are more equations than unknowns.

Therefore, a least squares solution to the linear system is used. The objective of linear least squares is to solve the following system for $\mathbf{x}$ [31]. Note that the following development of the linear (and weighted) least squares solution follows the development given in [31]. The development starts with

$$\mathbf{Ax} = \mathbf{\Omega} \tag{2.26}$$

where $\mathbf{A}$ is $m \times n$ where $m \geq n$, $\mathbf{x}$ is an $n \times 1$ vector, and $\mathbf{\Omega}$ is a $m \times 1$ vector. In the general case, where m may be greater than n with rank($\mathbf{A}$) = n this system is unsolvable. Therefore, the "closest" solution is found. The closest solution is defined as the solution which minimizes

$$E^2 = ||\mathbf{\Omega} - \mathbf{A}\mathbf{x}||^2 \tag{2.27}$$

where E is a value representing the error in a solution to Equation (2.26).

The value of $\mathbf{x}$ that minimizes E^2 is the least squares solution to Equation (2.26). From [31], the least squares solution $\hat{\mathbf{x}}$ is the solution to

$$\mathbf{A}^\top \mathbf{A}\hat{\mathbf{x}} = \mathbf{A}^\top \mathbf{\Omega} \tag{2.28}$$

Note that $\mathbf{A}^\top \mathbf{A}$ is invertible if and only if the columns of $\mathbf{A}$ are linearly independent. This is true for most cases involving random data, especially if $m >> n$. Therefore, the least squares solution $\hat{\mathbf{x}}$ is

$$\hat{\mathbf{x}} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{\Omega} \tag{2.29}$$

Some systems of equations may be attenuated by random noise. In this case, the equations in the system may not be equally reliable. For this situation, the best solution to Equation (2.26) is the solution that is most likely to minimize

$$E^2 = ||\mathbf{W}\mathbf{\Omega} - \mathbf{W}\mathbf{A}\mathbf{x}||^2 \tag{2.30}$$

where $\mathbf{W}$ is a matrix of weights for each equation. This solution is known as the weighted least squares solution $\mathbf{x_W}$. It is the solution to

$$\mathbf{W}\mathbf{A}\hat{\mathbf{x}}_\mathbf{W} = \mathbf{W}\mathbf{\Omega} \tag{2.31}$$

The solution to this equation is given by

$$\begin{aligned}(\mathbf{WA})^\top \mathbf{WA}\hat{\mathbf{x}}_\mathbf{W} &= (\mathbf{WA})^\top \mathbf{W\Omega} \\ \mathbf{A}^\top \mathbf{W}^\top \mathbf{WA}\hat{\mathbf{x}}_\mathbf{W} &= \mathbf{A}^\top \mathbf{W}^\top \mathbf{W\Omega} \\ \hat{\mathbf{x}}_\mathbf{W} &= (\mathbf{A}^\top \mathbf{W}^\top \mathbf{WA})^{-1} \mathbf{A}^\top \mathbf{W}^\top \mathbf{W\Omega}. \end{aligned} \tag{2.32}$$

Notice that the symmetric matrix $\mathbf{W}^\top\mathbf{W}$ appears twice in Equation (2.32). This matrix has an important stochastic interpretation. Suppose that the elements of $\mathbf{\Omega}$ are normally distributed. In this case,

$$\mathbf{\Sigma}^{-1} = \mathbf{W}^\top \mathbf{W} \tag{2.33}$$

where $\mathbf{\Sigma}^{-1}$ is the inverse of the covariance matrix $\mathbf{\Sigma}$ of $\mathbf{\Omega}$ [8], [31]. The covariance matrix is of the form

$$\mathbf{\Sigma}_{ij} = \sigma_i \sigma_j \tag{2.34}$$

where σ_i^2 is the variance in the i^{th} equation of Equation (2.31). Note that some of the randomly attenuated equations may be interrelated. Random attenuation in one equation may affect another. This is accounted for by the covariance terms of $\mathbf{\Sigma}$. The covariance terms are $\mathbf{\Sigma}_{ij}$, $i \neq j$.

Therefore, in the presence of a normally distributed $\mathbf{\Omega}$, Equation (2.32) may be reduced to

$$\hat{\mathbf{x}}_\mathbf{W} = (\mathbf{A}^\top \mathbf{\Sigma}^{-1} \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{\Sigma}^{-1} \mathbf{\Omega} \tag{2.35}$$

This equation gives the weighted least squares solution to Equation (2.31). Given this discussion, the nonlinear weighted least squares problem that the NLO solves may be developed.

2.4.3 Weighted Nonlinear Least Squares.

NLO is a method for approximating the weighted nonlinear least square solution to a system of nonlinear equations. Let $\mathbf{\Omega}(\mathbf{x})$ be an $m \times 1$ vector of m nonlinear functions where

$$\mathbf{\Omega}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}_1) & f_2(\mathbf{x}_2) & \dots & f_m(\mathbf{x}_m) \end{bmatrix}^\top \tag{2.36}$$

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_m \end{bmatrix}^\top \tag{2.37}$$

In this problem, a desired output $\mathbf{\Omega}(\mathbf{x}_d)$ is known. A particular solution x_d is desired given by

$$\{\mathbf{x} | \mathbf{\Omega}(\mathbf{x}) = \mathbf{\Omega}(\mathbf{x}_d)\} \tag{2.38}$$

where $\mathbf{x}_d$ is the input vector that produces the desired output $\mathbf{\Omega}(\mathbf{x}_d)$. Compare Equation (2.38) to Equation (2.22). These are very similar except that $\mathbf{\Omega}(\mathbf{x})$ represents a set of functions.

Because the system of nonlinear equations may be overdetermined, there may not be a solution to Equation (2.38). Even if the system were exactly determined, an analytical solution may be intractable. Therefore, the objective is to find the weighted least squares solution. However, since the system of equations is nonlinear, even the weighted least squares solution cannot be analytically calculated as in Equation (2.35). The Gauss-Newton method (which is called NLO in this thesis due to the application) may be employed to resolve this difficulty [2].

In the region about a some initial guess of $\mathbf{x}_d$ called $\mathbf{x}_0$, it is assumed that

$$\mathbf{\Omega}(\mathbf{x}) \approx \mathbf{\Omega}(\mathbf{x}_0) + \mathbf{J}(\mathbf{x})(\mathbf{x} - \mathbf{x}_0) \tag{2.39}$$

where $\mathbf{J}(\mathbf{x})$ is a matrix known as the Jacobian [2]. The Jacobian is defined as

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} (\nabla f_1(\mathbf{x}))^\top \\ (\nabla f_2(\mathbf{x}))^\top \\ \vdots \\ (\nabla f_m(\mathbf{x}))^\top \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_m} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{x})}{\partial x_m} \\ \vdots & & & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \frac{\partial f_m(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_m} \end{bmatrix} \tag{2.40}$$

Next, Equation (2.39) is expressed in an intuitive form similar to that given in Equation (2.24), ie

$$\Delta\boldsymbol{\Omega} = \mathbf{J}(\mathbf{x}_0)(\mathbf{x}_d - \mathbf{x}_0) \tag{2.41}$$

where $\Delta\boldsymbol{\Omega}(\mathbf{x})$ is the residual. To make the equation's intuitive nature clear, let $\Delta\mathbf{x} = \mathbf{x}_d - \mathbf{x}_0$, so the equation is expanded as

$$\begin{bmatrix} \Delta f_1 \\ \Delta f_2 \\ \vdots \\ \Delta f_m \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x}_0)}{\partial x_1}\Delta\mathbf{x}_1 + \frac{\partial f_1(\mathbf{x}_0)}{\partial x_2}\Delta\mathbf{x}_2 + \cdots + \frac{\partial f_1(\mathbf{x}_0)}{\partial x_m}\Delta\mathbf{x}_m \\ \frac{\partial f_2(\mathbf{x}_0)}{\partial x_1}\Delta\mathbf{x}_1 + \frac{\partial f_2(\mathbf{x}_0)}{\partial x_2}\Delta\mathbf{x}_2 + \cdots + \frac{\partial f_2(\mathbf{x}_0)}{\partial x_m}\Delta\mathbf{x}_m \\ \vdots \\ \frac{\partial f_m(\mathbf{x}_0)}{\partial x_1}\Delta\mathbf{x}_1 + \frac{\partial f_m(\mathbf{x}_0)}{\partial x_2}\Delta\mathbf{x}_2 + \cdots + \frac{\partial f_m(\mathbf{x}_0)}{\partial x_m}\Delta\mathbf{x}_m \end{bmatrix} \tag{2.42}$$

From this equation, it can be seen that the Jacobian is analogous to the derivative of $f(x)$ in Equation (2.23).

Similarly to Newton's method, the objective is to solve Equation (2.41) for $\mathbf{x}_d$. $\mathbf{J}(\mathbf{x}_0)$ is an $m \times n$ matrix where $m \geq n$. Often, this system is overdetermined and unsolvable. Also note that weighting has not been included. Therefore, the weighted linear least squares solution $\hat{\mathbf{x}}_d$ to Equation (2.41) is the solution to

$$\mathbf{W}\mathbf{J}(\mathbf{x}_0)(\hat{\mathbf{x}}_d - \mathbf{x}_0) = \mathbf{W}\Delta\boldsymbol{\Omega} \tag{2.43}$$

The solution to this equation for $\hat{\mathbf{x}}_d$ as given by Equation (2.35) is

$$\hat{\mathbf{x}}_d = \mathbf{x}_0 + (\mathbf{J}(\mathbf{x}_0)^\top \boldsymbol{\Sigma}^{-1} \mathbf{J}(\mathbf{x}_0))^{-1} \mathbf{J}(\mathbf{x}_0)^\top \boldsymbol{\Sigma}^{-1} \Delta\boldsymbol{\Omega} \tag{2.44}$$

As in Newton's method, the value of $\hat{\mathbf{x}}_d$ should produce a value of $\boldsymbol{\Omega}(\hat{\mathbf{x}}_d)$ that is closer to the desired output $\boldsymbol{\Omega}(\mathbf{x}_d)$. Therefore, $\hat{\mathbf{x}}_d$ becomes the guess for the second iteration of the method by letting $\mathbf{x}_1 = \hat{\mathbf{x}}_d$. The process then repeats. After multiple iterations, the value of $\hat{\mathbf{x}}_d$ converges to the weighted least squares solution of Equation (2.38). The process halts once the residual reaches a value specified by the user.

Recall from Section 2.4.2 that in the presence of Gaussian noise, $\boldsymbol{\Sigma}$ is the covariance matrix of that distribution. Furthermore, it is shown in [18] that $\hat{\mathbf{x}}_d$ converges to the local maximum likelihood estimate. That is, $\hat{\mathbf{x}}_d$ converges to the most probable $\mathbf{x}$ given $\boldsymbol{\Omega}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}$. The remaining task is finding a means of describing the confidence in $\hat{\mathbf{x}}_d$.

Fortunately this method naturally produces a means of computing the confidence in $\hat{\mathbf{x}}_d$. The next section describes how NLO is used to calculate the confidence and produce an associated visualization.

2.5 Estimating and Visualizing Confidence

Sections 2.3 and 2.4 provide the basis for the geolocation algorithms developed in this thesis; however, the visualization scheme has not been addressed. The objective of this section is to develop a method for visualizing the confidence in an estimated parameter $\hat{\mathbf{x}}$. This will be accomplished with the covariance matrix of $\hat{\mathbf{x}}$ called $\boldsymbol{\Sigma}_{\mathbf{x}}$.

The covariance matrix describes the variance and covariance in $\hat{\mathbf{x}}$. The covariance matrix needs to be calculable and put into a visually intuitive form. The visualization scheme will be given first, followed by an explanation of how $\boldsymbol{\Sigma}_{\mathbf{x}}$ is calculated.

A method for creating this visualization is found in [3]. This visualization method begins by calculating the Cholesky Decomposition of the covariance matrix. It decomposes

a matrix $\mathbf{\Sigma_x}$ as [31]

$$\mathbf{\Sigma_x} = LL^* \tag{2.45}$$

where L is a lower triangular matrix. Note that the Cholesky Decomposition only operates on Hermitian positive-definite matrices [3]. Fortunately, all covariance matrices are Hermitian and positive semi-definite [31]. In this way, L is analogous to the square-root of $\mathbf{\Sigma_x}$.

Using L, it is possible to generate samples of $\mathbf{x}$ that follow a distribution with a covariance matrix of $\mathbf{\Sigma_x}$ from normally distributed samples $\mathbf{y}$ as given by

$$\begin{aligned} &\mathbf{Y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ &\hat{\mathbf{x}} = L\mathbf{Y} \\ &\quad \rightarrow \hat{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma_x}) \end{aligned} \tag{2.46}$$

This procedure will create a scatter of $\hat{\mathbf{x}}$ values. However, if rather than pulling $\mathbf{Y}$ values from a normal distribution, they are samples of a sphere of radius σ, then $\hat{\mathbf{x}}$ will sample the contour of its distribution associated with σ. The contours of $\mathbf{\Sigma_x}$ are ellipsoids [2]. Therefore, the process for visualizing these ellipsoids is given by

$$\begin{aligned} &\mathbf{Y} \sim \text{Sphere(Radius} = \sigma) \\ &\hat{\mathbf{x}} = L\mathbf{Y} \\ &\quad \rightarrow \text{Plot Surface}(\hat{\mathbf{x}}) \end{aligned}$$

where σ is the standard deviation associated with the ellipsoid.

The standard deviation relates to a percent probability that the samples $\hat{\mathbf{x}}$ should exist inside of the ellipsoid. Thus the percent probability p_σ that $\mathbf{x}$ will exist inside of an ellipsoid

generated with a given σ may be found. This ellipse is what is referred to in this thesis as the error surface. It should be noted that the equation relating p_σ and σ depends on the dimension of $\mathbf{\Sigma_x}$. It is commonly known that in a 1-D normal distribution that 3σ corresponds to a 99% probability. This is not true for a 3-D normal distribution. For a 3-D normal distribution, 3σ corresponds to a probability of 97%.

This procedure provides a method for creating a visualization of the covariance matrix; however, it does not provide a means of finding the covariance matrix. Fortunately, the covariance matrix can easily be produced for the NLO. The following development closely follows that given in [3]; however, it has been tailored to this application. First the definition of $\mathbf{\Sigma_x}$ is given by

$$\mathbf{\Sigma_x} \triangleq E[(\mathbf{x}_d - \hat{\mathbf{x}}_d)(\mathbf{x}_d - \hat{\mathbf{x}}_d)^\top] \tag{2.47}$$

where $\mathbf{x}_d$ is the value of $\mathbf{x}$ that solves Equation (2.38) exactly. Recall that the reason $\mathbf{x}_d$ cannot be exactly found is because of the random noise in the system. In Equation (2.47), $\hat{\mathbf{x}}_d$ is a random variable representing the weighted nonlinear least squares solution given by Equation (2.44). Next, let the noise in the system be described by a random variable $\boldsymbol{\nu}$ with the properties

$$E(\boldsymbol{\nu}) = \mathbf{0} \tag{2.48}$$

$$E(\boldsymbol{\nu}\boldsymbol{\nu}^\top) = \mathbf{\Sigma_\Omega}. \tag{2.49}$$

Using this description of the noise in the system and ignoring the shift $\mathbf{x}_0$, the left hand side of Equation (2.43) is rewritten as

$$\mathbf{J}^\top\mathbf{W}^\top\mathbf{W}\mathbf{J}\hat{\mathbf{x}}_d = \mathbf{J}^\top\mathbf{W}^\top\mathbf{W}\mathbf{J}\mathbf{x}_d + \mathbf{J}^\top\mathbf{W}^\top\mathbf{W}\boldsymbol{\nu}. \tag{2.50}$$

The only random attenuation in the system is due to $\boldsymbol{\nu}$. Therefore, the covariance matrix is $\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1} = \mathbf{W}^{\top}\mathbf{W}$. Also keep in mind in this development that $\hat{\mathbf{x}}_d$ and $\mathbf{x}_d$ are random variables. Equation (2.50) is then simplified as

$$\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}(\hat{\mathbf{x}}_d - \mathbf{x}_d) = \mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\boldsymbol{\nu}. \tag{2.51}$$

This equation will be used to find a simple means of computing $\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}$ in terms of known values. $\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}$ is found by manipulating both sides of Equation (2.51), ie

$$\begin{aligned}
E\left[\left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}(\hat{\mathbf{x}}_d - \mathbf{x}_d)\right)\left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}(\hat{\mathbf{x}}_d - \mathbf{x}_d)\right)^{\top}\right] &= E\left[\left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\boldsymbol{\nu}\right)\left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\boldsymbol{\nu}\right)^{\top}\right] \\
E\left[\left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}(\hat{\mathbf{x}}_d - \mathbf{x}_d)\right)\left((\hat{\mathbf{x}}_d - \mathbf{x}_d)^{\top}\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}\right)\right] &= E\left[\left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\boldsymbol{\nu}\right)\left(\boldsymbol{\nu}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}\right)\right] \\
\left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}\right)E\left[(\hat{\mathbf{x}}_d - \mathbf{x}_d)(\hat{\mathbf{x}}_d - \mathbf{x}_d)^{\top}\right]\left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}\right) &= \mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}E\left[\boldsymbol{\nu}\boldsymbol{\nu}^{\top}\right]\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J} \\
\left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}\right)E\left[(\hat{\mathbf{x}}_d - \mathbf{x}_d)(\hat{\mathbf{x}}_d - \mathbf{x}_d)^{\top}\right]\left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}\right) &= \mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J} \\
\boldsymbol{\Sigma}_{\mathbf{x}}\left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}\right) &= \mathbf{I} \\
\boldsymbol{\Sigma}_{\mathbf{x}} &= \left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}\right)^{-1}
\end{aligned} \tag{2.52}$$

This equation provides a simple means of calculating $\boldsymbol{\Sigma}_{\mathbf{x}}$ from the known matrices $\mathbf{J}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}$. Next, the location of the error surface in space must be determined.

To this end, it will be demonstrated that $\hat{\mathbf{x}}_d$ is an unbiased estimate of $\mathbf{x}_d$. By taking the expected value of both sides of Equation (2.51) we have

$$\begin{aligned}
E\left[\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}(\hat{\mathbf{x}}_d - \mathbf{x}_d)\right] &= E\left[\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\boldsymbol{\nu}\right] \\
\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}(E\left[\hat{\mathbf{x}}_d\right] - \mathbf{x}_d) &= \mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}E\left[\boldsymbol{\nu}\right] \\
\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}(E\left[\hat{\mathbf{x}}_d\right] - \mathbf{x}_d) &= \mathbf{0} \qquad \text{from Equation (2.48)} \\
\mathbf{x}_d &= E\left[\hat{\mathbf{x}}_d\right]
\end{aligned} \tag{2.53}$$

This final equation follows since it is assumed that $\mathbf{J}^\top \boldsymbol{\Sigma}_{\boldsymbol{\Omega}} \mathbf{J}$ is invertible and therefore has no null space [31]. This demonstrates that the distribution of $\hat{\mathbf{x}}_d$ is centered on $\mathbf{x}_d$ and thus is an unbiased estimate.

So far, this visualization method is only applicable to the NLO because no method has been provided to calculate $\boldsymbol{\Sigma}_{\mathbf{x}}$ for the TA. Moreover, no method has been given to evaluate if the covariance matrix of $\hat{\mathbf{x}}_d$ accurately estimates the confidence in $\hat{\mathbf{x}}_d$. These issues are addressed in the next section.

2.6 Monte Carlo Simulations

In this section, the theory and utility of Monte Carlo Simulations (MCS) are discussed. MCS are investigated here as a means of providing the confidence surface for the TA and the NLO. The confidence surface for the NLO produced by MCS is treated as truth for comparison against the confidence surface produced via Equation (2.52). The outputs of some systems operating on random variables may not be analytically determined, but MCS provides a means of statistical inference from simulations [12], [28].

The fundamental idea behind MCS is to determine the distribution of a system's output from many simulated inputs produced from the inputs' distributions. In our case, for a given sensor configuration, LOS measurements are simulated randomly according to their distributions. Each set of random measurements is used to estimate the subject. This produces a distribution of subject states. This method is illustrated in Figure 2.7.

The process begins by using the N sensors' measurements as the means of their respective distributions. The random measurements resulting from the sensors' distributions are used to produce geolocation estimate. This process is repeated until M estimates are produced. The geolocation covariance matrix is estimated from the M geolocation estimates.

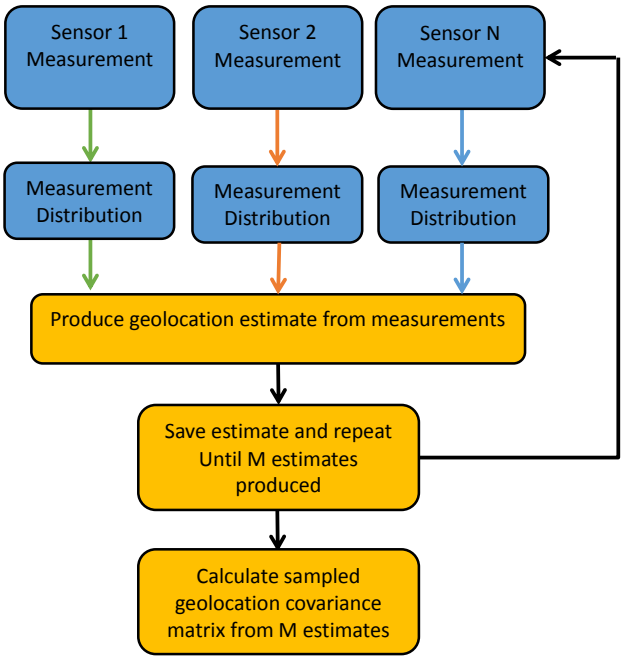

Figure 2.7: Monte Carlo Simulation covariance matrix estimation.

The MCS produces a scatter of geolocation estimates. An example distribution of position estimates is shown in Figure 2.8. Since the sensors' measurements are used as the means of their distributions, the scatter of geolocation estimates produced by the MCS will be centered on the estimate given by the sensor measurements. Thus the distribution will have the same mean as the NLO.

The primary purpose of producing MCS is to provide truth for the confidence surface provided by the NLO. To accomplish this task, the distribution of the scatter is described in terms of a covariance matrix.

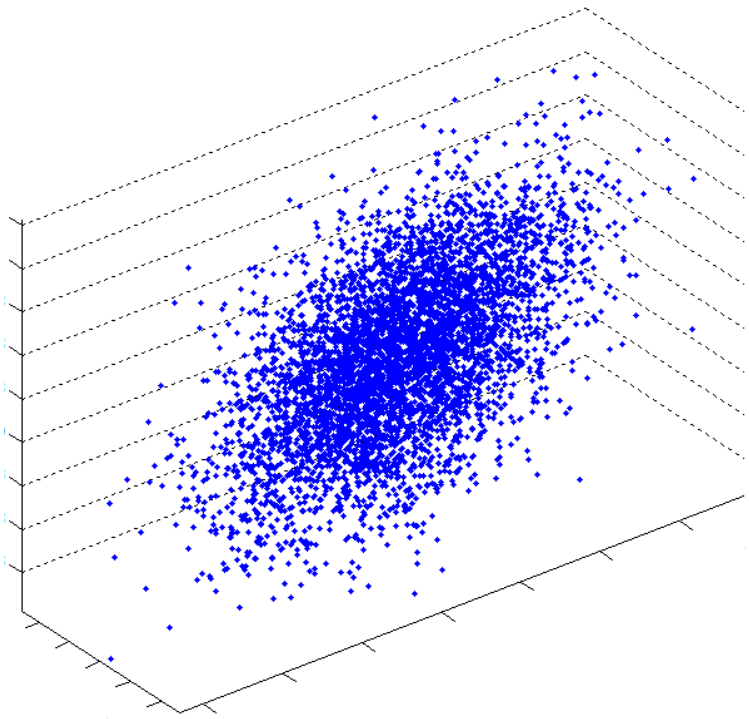

Figure 2.8: Scatter of geolocation position estimates from MCS.

Let the number of MCS estimates be M. The i^{th} estimate is $\mathbf{x}_i$. The sample covariance matrix may be calculated as [13]

$$\mathbf{\Sigma}_{\mathbf{x}} = \frac{1}{M-1} \sum_{i=1}^{M} \left(\mathbf{x}_i - E\left[\mathbf{x}_i\right]\right) \left(\mathbf{x}_i - E\left[\mathbf{x}_i\right]\right)^{\top}. \tag{2.54}$$

For a large value of M, the sample covariance matrix will converge to the true covariance matrix. Therefore, the sample covariance matrix is an estimate of the true covariance matrix. Observe that in Figure 2.7, the method for producing geolocation estimates from measurements is not given. MCS can be used to produce an estimate of the true covariance matrix for any geolocation algorithm. Therefore, MCS provides a means of estimating the covariance matrix for the TA and NLO.

Note that the value of M, for which the sample covariance matrix converges sufficiently, is unknown. One method of choosing M is to select a value that from

experience has been shown to converge. Another method is to keep increasing the number of MCS samples until the sample covariance matrix changes by a sufficiently smaller amount. This provides a method for visualizing the confidence estimates for the geolocation algorithms.

MCS is the only means identified presently for accurately producing the covariance matrix for the TA. It also provides the ability to check the accuracy of the covariance matrix produced by the NLO.

The application of these algorithms on simulated data is described in the following chapter. The methodology addresses how the TA and NLO are compared against each other and how the NLO may be improved to take into account more sources of error. These and other practical concerns regarding the development of this geolocation method and confidence estimation and visualization scheme are described in the following chapter.

III. Methodology

This chapter discusses the methodology for evaluating the efficacy of Non-Linear Optimization (NLO) and determining which NLO(s) provides the best performance. In Chapter 1, the TA is described as a standard means of estimating the subject's location from Line-of-Sight (LOS) measurements. The literature review in Chapter 2 led to interest in NLO, otherwise known as the Gauss-Newton method. The objective of this research is to compare the performance of the NLO with the TA, and the benefits provided by improvements to the NLO. Furthermore, it is desirable for system operators to be capable of visualizing the confidence in geolocation algorithms. Therefore, a comparison of confidence estimates is another research objective.

The methodology used in this research is to simulate scenarios, generate measurements, and run the various algorithms on this simulated data. The first stage in this method is simulating the scenario. The subject being localized is used to parameterize the sensor generation. Therefore, the flightpath generation is described first.

3.1 System Geometry Generation

3.1.1 Time Generation.

The measurement times are the first part of the scenario simulation. The times are used to generate the subject's flightpath and the sensor positions. Therefore, timing generation is the first stage in generation of the system's geometry (that is, the positions of the subject and the observing sensors).

Sensors may produce LOS measurements of the subject at different times. Moreover, it may be desirable to use sensor measurements that are close to the time at which the user would like to geolocate the subject.

To this end, measurement times are generated in a pseudorandom manner. The measurement time for each sensor's initial measurement time t_0 is chosen according to a uniform distribution. The parameters of the uniform distribution were arbitrarily chosen as

$$t_0 \sim Unif\,(1, 9)\ \text{s} \tag{3.1}$$

Subsequent measurement times t_k are generated in the same manner, using the previous time as a starting point as

$$t_k \sim t_{k-1} + Unif\,(1, 9)\ \text{s} \tag{3.2}$$

This process is performed for each sensor until t_k becomes larger than a threshold $t_{\max}$. Sensor measurements and truth data (the subject's true positions and/or velocities) are produced for these times. The time values determine when the subject's path is sampled. In the next section, the synthesis of the SV's flightpath is described.

3.1.2 Space Vehicle Generation.

Given that the performance of geolocation algorithms depends on the flightpath of the subject under observation, this research is simplified by limiting the subject to one type. The most challenging subject, the Space Vehicle (SV), is used to evaluate the performance of the NLOs and TA. The model for the SV, detailed in Section 2.2, contains sinusoidal elements. Sinusoids are infinitely differentiable, so the flightpath contains an infinite number of moments. Furthermore, the tremendous speed of an SV exacerbates the sources of error discussed in Chapter 1. Therefore, an SV's flightpath seems to encapsulate the greatest extent of the sources of geolocation error.

The flightpath is pseudorandomly generated by randomly producing a few of the flightpath parameters from Section 2.2. These values allow the remaining parameters to be calculated.

The orientation of the SV's flightpath is determined by randomly generating a unit-vector **n** as given by

$$\tilde{\mathbf{n}} \sim \mathcal{N}(0, \mathbf{I}_3) \tag{3.3}$$

$$\mathbf{n} = \frac{\tilde{\mathbf{n}}}{||\tilde{\mathbf{n}}||}. \tag{3.4}$$

This vector is orthogonal to every vector contained in a plane. The flightpath must be contained in a plane perpendicular to this plane. This setup is shown in Figure 3.1. There are an infinite number of planes orthogonal to the plane shown in blue in Figure 3.1. The plane containing the flightpath is determined via a uniformly random rotation of the flightpath about **n**. The MATLAB® function used to simulate the satellites is given in Appendix A. The shape of the flightpath is created by randomly generating four of its parameters.

The arc-angle (γ in Figure 2.3), apogee, and final altitude a_f and speed at the end of the reentry phase v_f are randomly generated as

$$\gamma \sim Unif\left(5.5 \times 10^6/r, \pi\right) \text{ Radians} \tag{3.5}$$

$$\text{apogee} \sim Unif\left(900 \times 10^3, 1.5 \times 10^6\right) \text{ m} \tag{3.6}$$

$$a_f \sim Unif\left(150 \times 10^3, 450 \times 10^3\right) \text{ m} \tag{3.7}$$

$$v_f \sim Unif\left(5 \times 10^3, 7 \times 10^3\right) \text{ m/s}. \tag{3.8}$$

Equation (3.5) is an estimate of the typical ranges these values would have for a Low-Earth Orbit (LEO) satellite. The remaining three parameters are the result of

testing which values produce reasonable results. The approximate orbital speed of a LEO satellite is 7.6 km/s. Therefore, the velocity at the transition to the reentry phase is randomly produced as given by Equation (3.8). Its apogee is approximately 1,120 km. Therefore, Equation (3.6) keeps the apogee near this value. Equation (3.7) and the others in combination are used to ensure that the flightpath is never inside the Earth. The result of this SV generation effort is given in Chapter 4.

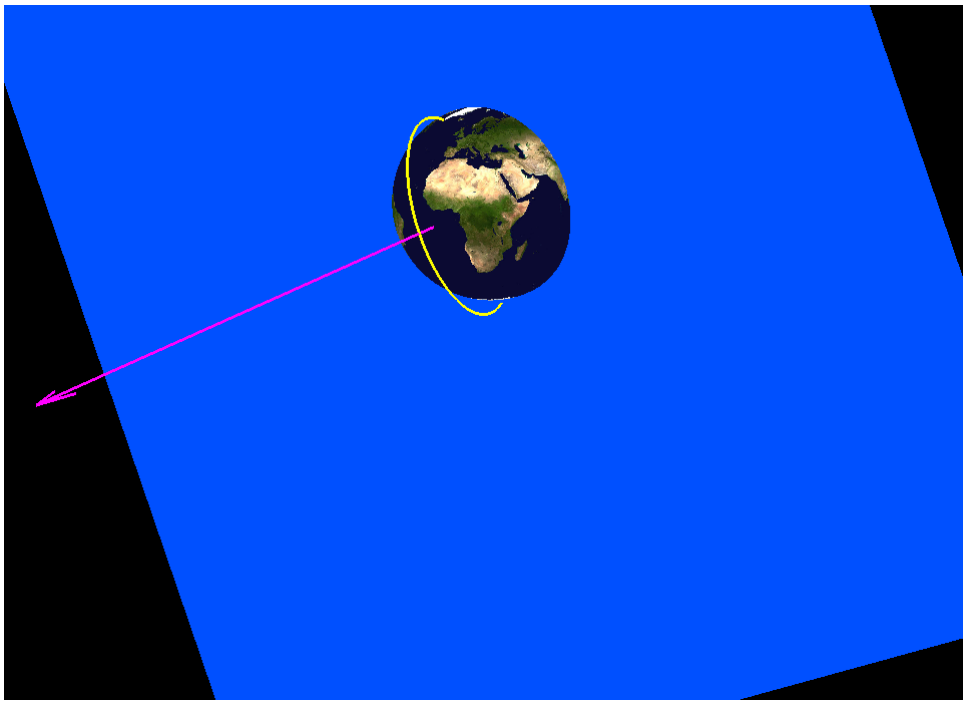

Figure 3.1: SV vehicle flightpath generation. An SV flightpath is shown in yellow. The vector **n** that determines the orientation of the flightpath is shown in magenta. The plane orthogonal to the plane containing the flightpath is shown in blue.

Once the flightpath has been generated, positions and velocities of the SV along the flightpath may be calculated. Positions and velocities are calculated at each time that a measurement is taken by a sensor. This is done to provide simulated truth data. In this way, the estimated SV state at these times may be compared to its estimated state. The positions are calculated according to Equation (2.19). The velocity is calculated via the

first derivative of Equation (2.19). The next section will describe how the SV's flightpath is used to parameterize the satellite generation.

3.1.3 Satellite Generation.

Though the algorithms in question may be implemented on data from any type of sensor that measures LOS, satellites are simulated for this research. Satellites are chosen because satellites measure the LOS to the subject across the largest distances. However, some satellites may also be very close to the subject because various SV's may approach the lower satellite orbits. Thus, out of all sensor platforms, satellites provide the most diversity in range to the subject. The first step in the satellite generation is the orbit generation. The code that performs the satellite generation is given in Appendix A.

Four typical satellites orbits were considered: Low Earth Orbit (LEO), Medium Earth Orbit (MEO), Geosynchronous Earth Orbit (GEO), and High Earth Orbit (HEO). There are many satellites in various versions of these orbits. Therefore, a general literature review was performed to develop approximate models of these orbits. While the models may not be precise, the most important feature of the orbits for this research is their orbital distance. The models for these orbits provide the desired diversity in the oribtal distances. The model of each satellite is given in the orbital generation function OrbGen.m shown in Appendix A. The orientations of the orbits are chosen pseudorandomly.

The orbits are generated at random with the constraint that all satellites must have LOS to the SV. In addition to realism, this constraint was added to prevent satellites from observing a subject over unrealistic distances. For example, if a LEO satellite is on the opposite side of the Earth relative to the observed subject, the error in its LOS propagates over a larger distance than would otherwise be possible. This constraint is implemented by taking advantage of the randomly generated vector $\mathbf{n}$ from the previous section. The constraint is illustrated in Figure 3.2.

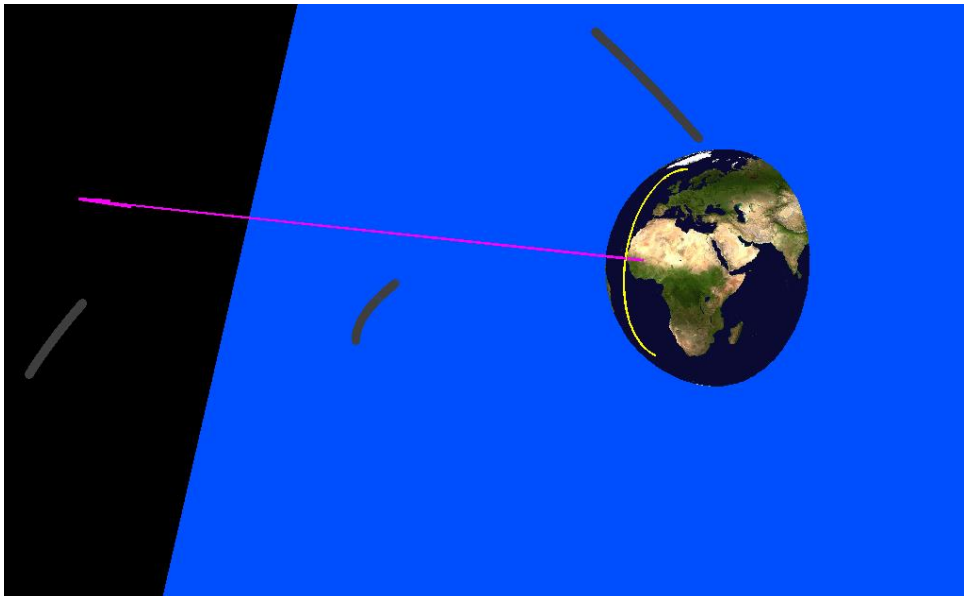

Figure 3.2: Satellite orbit synthesis. As before, the magenta line represents **n**. The restriction on the satellite's orbits is that they must be on the same side of the blue plane as the SV.

A simple means of ensuring that each satellite has LOS to the subject at the time of each measurement was not found. Moreover, this requirement is unnecessary. The purpose of this requirement is to ensure that sensors do not observe the subject at absurd distances. Therefore, this requirement was relaxed and substituted for the requirement that the satellites and the subject are on the same side of the Earth. This process is tedious and is left to the code given in Appendix A. A notional explanation of the satellite generation is described below.

The satellite orbits are randomly generated. The length of time needed to observe the subject is used to calculate a section of the satellite's orbit over which the satellite observes the subject. This section is then rotated (if needed) until both the endpoints of the section are on the same side of the Earth as the subject. The sensors are on the same side of the Earth meaning that the satellites are on the same side of the plane whose span is orthogonal to **n**. This is the plane shown in blue in Figure 3.2. The satellite orbits and SV flightpath

form the basis for LOS generation. The method for simulating the LOS for each algorithm is described in the section 3.2.

3.2 LOS Generation

This section discusses how LOS measurements are simulated. LOS measurements are generated for three purposes. The first is to evaluate the comparative performance of the NLO to the TA. The second is verify the theoretical performance of the algorithms. Lastly, the LOS data is produced to determine the benefits provided from adding complexity to the NLO. Each NLO takes into account certain sources of error. The LOS synthesis for the Static NLO is described in the next section.

3.2.1 Stationary Subject.

Each NLO includes certain sources of LOS error in their model. The simplest NLO, the Static NLO, only takes into consideration the geometry of the system. It does not include sensor synchronization, the subject's velocity, and time delay in its model. Sensor synchronization means that the sensors obtain their measurements at the same time. The time delay is the time it takes for a signal coming from the subject to be received by the sensors.

The Static NLO assumes that the subject is stationary over the duration of the measurements used to provide an estimate. This assumption is inaccurate for fast moving subjects and unsynchronized sensor measurements. In such a case, the NLOs confidence surface is expected to be overly confident.

To verify that the static NLO performs correctly, a system is created where the sensors' measurements are perfectly synchronized, and the subject is stationary. The Static NLO is optimized for this type of scenario. If the static NLO has been correctly implemented, then its confidence estimate will be ideal and should match its MCS.

For the static NLO the LOS generation is straightforward. The measurement distribution follows from Figure 3.3. Since the subject is stationary, the ideal LOS

measurement is merely the difference vector between the subject and the sensor. Therefore, since it is assumed that the measurements have no bias, the means of the θ and ϕ distributions are the spherical angles associated with this difference vector. The difference vector $\boldsymbol{\psi} = \mathbf{x} - \mathbf{s}$. Given this difference vector and taking note of Equation (2.3), this means the distributions corresponding to the difference vector between the subject and a sensor can be calculated as

$$\begin{aligned} \mu_\theta &= \arctan 2\left(\boldsymbol{\psi}_y, \boldsymbol{\psi}_x\right) = \arctan 2\left(\mathbf{x}_y - \mathbf{s}_y, \mathbf{x}_x - \mathbf{s}_x\right) \\ \mu_\phi &= \arctan 2\left(\sqrt{\boldsymbol{\psi}_x^2 + \boldsymbol{\psi}_y^2}, \boldsymbol{\psi}_z\right) = \arctan 2\left(\sqrt{(\mathbf{x}_x - \mathbf{s}_x)^2 + (\mathbf{x}_y - \mathbf{s}_y)^2}, \mathbf{x}_z - \mathbf{s}_z\right) \end{aligned} \tag{3.9}$$

where μ_θ and μ_ϕ are the respective means of the θ and ϕ distributions. The distributions are assumed to be Gaussian as given by Equation (2.5). Because Equation (3.9) allows the means to be calculated, only the variances are unknown.

The variances of the θ and ϕ distributions are determined by the confidence in the LOS-measuring sensor. The variances are arbitrarily selected in this research. In the next section, this problem is expanded to include subject motion.

3.2.2 Adding Object Motion.

In this section, the LOS data takes into account linear motion. The subject has constant velocity, and the time delay is not included. The utility of this data is in verifying the functionality of the Velocity NLO. The Velocity NLO assumes that the subject moves in approximately a straight line over the duration of the measurements used to provide a geolocation estimate, and it does not include time delay in its model. Therefore, the Velocity NLO should converge to the ideal estimate of the subject's state and provide an ideal confidence surface.

Because the time delay is not included, the sensors measure the LOS to the subject instantaneously. Therefore, the means of the sensor distributions are calculated in the same

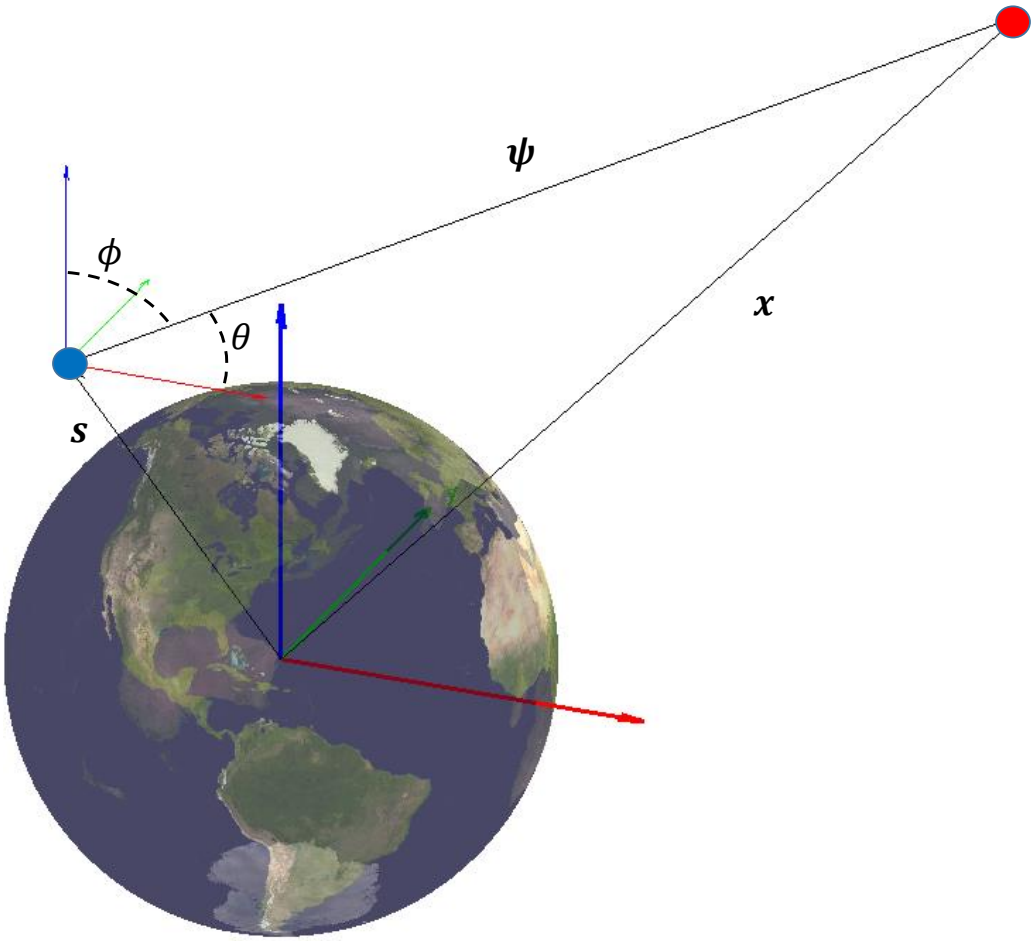

Figure 3.3: Static NLO LOS synthesis. The coordinates to the sensor are given by the vector **s**. The subject's coordinates are given by the vector **x**. $\boldsymbol{\psi}$ is the difference vector between the subject and the sensor. θ and ϕ are the spherical angles associated with $\boldsymbol{\psi}$.

manner as in the static case described by Equation (3.9). The only difference in this case is that the subject's position changes with time. The subject's position at a given time $\mathbf{p}(t)$ is described by the vector-valued function as given by

$$\mathbf{p}(t) = \mathbf{p}(t)|_{t=0} + \dot{\mathbf{p}}t. \tag{3.10}$$

In this equation, the subject's velocity is the first derivative of the position with respect to time and denoted by $\dot{\mathbf{p}}$. Note that the velocity is a constant and thus not a function of

time. Different velocities are used to test the NLOs. Next, the time delay is added to the linear model.

3.2.3 Adding Time Delay.

The LOS data generation scheme discussed in this section is used to verify the functionality of the most robust NLO: the time delay and velocity NLO. The LOS measurements described in this section include time delay for subject with a linear flightpath.

The time delay is particularly relevant for satellite systems. For example, a HEO satellite may have an approximate orbiting radius of 60×10^6 m. Radiation from a subject sitting on the Earth's surface requires approximately $(60 \times 10^6 - 6.731 \times 10^6)/3 \times 10^8$ s $\approx$ 200 ms to be received by a HEO sensor. Although small, this time delay may be significant for fast-moving subjects. The dead LEO satellite in Section 3.1.2 has a maximum speed of 7.6×10^3 m/s. In 200 ms, such a satellite could traverse 1.5 km! Thus, by the time the sensor has observed the subject, it has moved 1.5 km. The time delay means that sensors are not observing the subject where it currently is but where it used to be.

Therefore, the means of the θ and ϕ distributions are determined by the subject's position at the time that the sensor measures it minus the time delay. To solve this problem, the following definitions are made. The time delay is give the variable t_d. The time at which the subject is observed by a sensor, the measurement time, is called t_m. The LOS for a given sensor is then determined by the subject's position at $t_m - t_d$. This is concerned with data simulation. Therefore, the subject's motion is known. Its position at t_m is known, and the sensor's position is also known; therefore, finding the subject's position at $t_m - t_d$ requires knowledge of only t_d. A method for calculating t_d is developed using Figure 3.4.

The time delay is calculated by finding an equation for the time delay in terms of the the subject's position and velocity and the sensors position. Because this is the synthesis

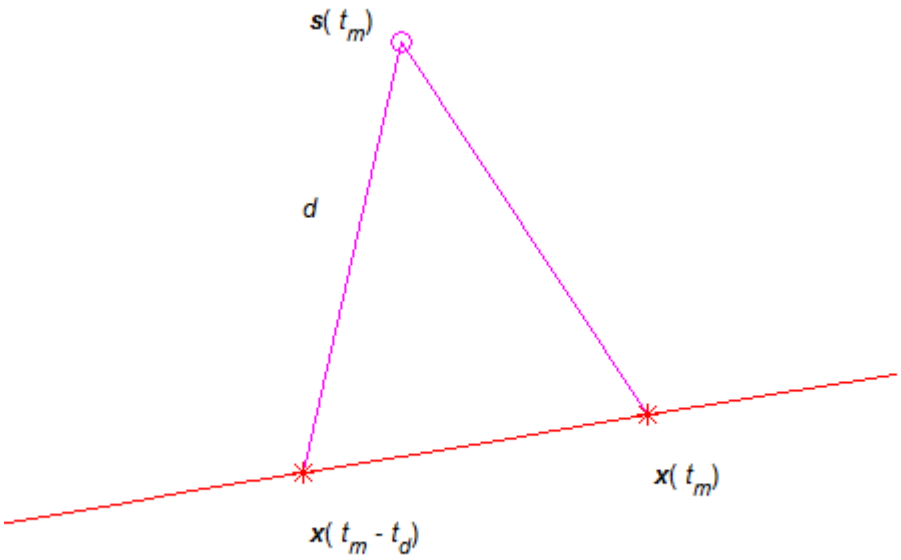

Figure 3.4: LOS with time delay synthesis setup. The sensors position at the measurement time t_m is given by $\mathbf{s}(t_m)$. Two of the subject's positions are shown as the asterisks. Its flightpath is shown as the straight lines through the asterisks. The distance between the sensor at time t_m and the subject at time $t_m - t_d$ is denoted by d.

problem, the subject's motion is known. Therefore,

$$\mathbf{x}(t_m - t_d) = \mathbf{x}(t_m) - \dot{\mathbf{x}}t_d \tag{3.11}$$

Because the subject's position and velocity are known, the unknowns in this equation are t_d and $\mathbf{x}(t_m - t_d)$. Another independent equation is developed to take advantage of the fact that the sensor's location is known, yielding an equation for t_d.

The time delay is the time it takes for the signal emitted from the subject at $t_m - t_d$ to reach the sensor at t_m. The signal traverses a distance of d at a rate of the speed of light c.

Therefore, from Figure 3.4,

$$d = ||\mathbf{x}(t_m - t_d) - \mathbf{s}(t_m)|| \tag{3.12}$$

Because the speed of light is a known constant, t_d may be given as

$$t_d = \frac{||\mathbf{x}(t_m - t_d) - \mathbf{s}(t_m)||}{c} \tag{3.13}$$

Substituting Equation (3.11) into the above equation and multiplying by c produces

$$ct_d = ||\mathbf{x}(t_m) - \dot{\mathbf{x}}t_d - \mathbf{s}(t_m)|| \tag{3.14}$$

The unknown in this equation is t_d. Squaring both sides of the above equation produces a quadratic equation for t_d. This equation may be solved for t_d. Because the equation is quadratic, two solutions exist; however, only one is realistic. By observing the nature of the solution, the correct equation for t_d is selected. One of the solutions is negative. The negative solution produces a value of $t_m - t_d$ which is greater than t_m. By the problem's setup, the value of t_d must be positive, so the negative value cannot be the solution.

Given t_d, the means of the θ and ϕ distributions may be calculated as

$$\begin{aligned}
\mu_\theta &= \arctan 2\left(\mathbf{x}_y(t_m - t_d) - \mathbf{s}_y(t_m), \mathbf{x}_x(t_m - t_d) - \mathbf{s}_x(t_m)\right) \\
\mu_\phi &= \arctan 2\left(\mathbf{x}_z(t_m - t_d) - \mathbf{s}_z(t_m),\ \sqrt{(\mathbf{x}_x(t_m - t_d) - \mathbf{s}_x(t_m))^2 + (\mathbf{x}_y(t_m - t_d) - \mathbf{s}_y(t_m))^2}\right).
\end{aligned} \tag{3.15}$$

With t_d known, $\mathbf{x}(t_m - t_d)$ is also known as per Equation (3.11).

The next section discuses the synthesis of LOS data for a model of the most challenging subject. The most challenging scenario for geolocation is considered to be the case of observing a fast-moving subject such as an SV.

3.2.4 Space Vehicle Scenario.

Sections 3.2.1 to 3.2.3 describe LOS measurement synthesis for unrealistic subjects that are stationary or have linear flightpaths. This section describes LOS data generation for the most challenging subject: the Space Vehicle (SV).

The time delay cannot be analytically calculated for the SV. Therefore, time delay is calculated as part of the subject's flightpath. This process is illustrated by Figure 3.5. The idea is to first generate the measurement times for the sensors as described in Section 3.1.1. Next, the SV's flightpath is generated. The time values and the SVs flightpath parameters are used to generate sensor positions. The time delay that would exist between the sensors and the subject at the time of measurement t_m is then calculated via

$$t_d = \frac{||\mathbf{x}(t_m) - \mathbf{s}(t_m)||}{c}. \tag{3.16}$$

The time delay for each measurement time is added to the measurement time. The SV flightpath is simulated at the time corresponding to the sensors' measurements. These positions are the locations of the SV at measurement times t_m. With the subject's position at the measurement times calculated, the LOS angles associated with these positions may be simulated. This concludes discussion on the synthesis of data for the most challenging scenario.

3.3 Static NLO

This section describes how the static NLO is constructed. For the static NLO, it is assumed that the subject is stationary while sensors produce the measurements used to geolocate it. As discussed in Section 2.4, at the core of this method is the linearity assumption.

This assumption is captured in the Jacobian of the measurements as a function of the subject's state $\mathbf{\Omega}(\mathbf{x})$. This function outputs the ideal (noiseless) measurements that are

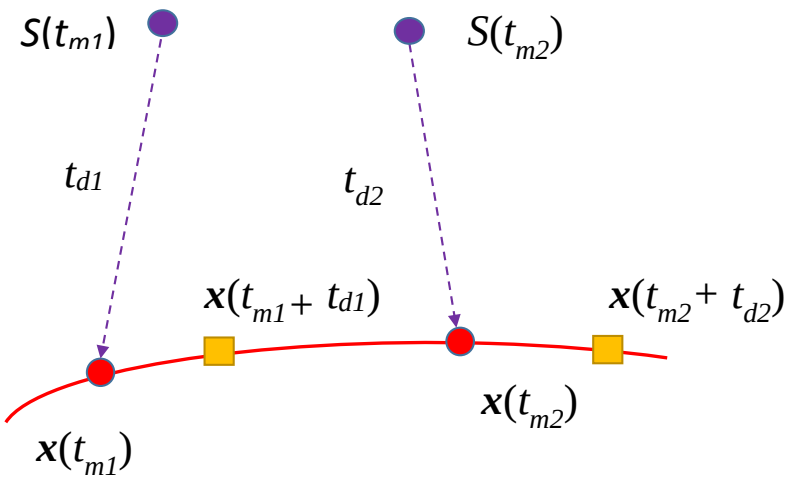

Figure 3.5: Geometry for space vehicle LOS synthesis. The subject's flightpath is shown as the curved line. Its positions at the measurement times are illustrated by the points along the path. The time delay associated with distance between the sensor at time t_m and the subject at the time of measurement t_m is denoted by t_d. The squares illustrate the SV's location at the t_m.

associated with the input subject state. Equation (3.9) may be used to find this function for one pair of θ and ϕ measurements in the static scenario (the subject is stationary). This equation gives the means of the θ and ϕ distributions, respectively. The means of these angles are the angles that should be measured in the absence of noise. Therefore, the ideal angles that should be measured for the static scenario may be calculated as

$$\begin{aligned} \boldsymbol{\psi} &= \mathbf{x} - \mathbf{s} \\ \theta &= \arctan 2\left(\boldsymbol{\psi}_y, \boldsymbol{\psi}_x\right) \\ \phi &= \arctan 2\left(\sqrt{\boldsymbol{\psi}_x^2 + \boldsymbol{\psi}_y^2}, \boldsymbol{\psi}_z\right) \end{aligned} \tag{3.17}$$

where $\mathbf{s}$ is a vector representing the sensor's position.

To localize the subject, several measurements must be used. Therefore, the Jacobian will be the concatenation of the Jacobian of each measurement as described in

Equation (2.39). Thus the Jacobian at $k-1$ is given as

$$\mathbf{J}_{k-1} = \begin{bmatrix} \frac{\partial\theta_1}{\partial\mathbf{x}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_1}{\partial\mathbf{x}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_1}{\partial\mathbf{x}_z}\Big|_{\mathbf{x}_{k-1}} \\ \frac{\partial\phi_1}{\partial\mathbf{x}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_1}{\partial\mathbf{x}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_1}{\partial\mathbf{x}_z}\Big|_{\mathbf{x}_{k-1}} \\ \frac{\partial\theta_2}{\partial\mathbf{x}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_2}{\partial\mathbf{x}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_2}{\partial\mathbf{x}_z}\Big|_{\mathbf{x}_{k-1}} \\ \frac{\partial\phi_2}{\partial\mathbf{x}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_2}{\partial\mathbf{x}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_2}{\partial\mathbf{x}_z}\Big|_{\mathbf{x}_{k-1}} \\ & \vdots & \\ \frac{\partial\theta_m}{\partial\mathbf{x}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_m}{\partial\mathbf{x}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_m}{\partial\mathbf{x}_z}\Big|_{\mathbf{x}_{k-1}} \\ \frac{\partial\phi_m}{\partial\mathbf{x}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_m}{\partial\mathbf{x}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_m}{\partial\mathbf{x}_z}\Big|_{\mathbf{x}_{k-1}} \end{bmatrix} \tag{3.18}$$

where each angle's subscript indicates the measurement with which it is associated. The partial derivatives from which the Jacobian is constructed are the partial derivatives of θ and ϕ as given in Equation (3.17). The values of these derivatives at $\mathbf{x}_{k-1}$ are not equal for different measurements because the sensor location is different for each measurement. These derivatives are given compactly in the form of the gradients of the measurement angles, ie

$$\begin{aligned} ||\boldsymbol{\psi}||^2 &= (\mathbf{x}_x - \mathbf{s}_x)^2 + (\mathbf{x}_y - \mathbf{s}_y)^2 + (\mathbf{x}_z - \mathbf{s}_z)^2 \\ ||\boldsymbol{\psi}_{xy}||^2 &= (\mathbf{x}_x - \mathbf{s}_x)^2 + (\mathbf{x}_y - \mathbf{s}_y)^2 \\ \nabla_{\mathbf{x}}\theta &= \begin{bmatrix} -\frac{\mathbf{x}_y - \mathbf{s}_y}{||\boldsymbol{\psi}_{xy}||^2} & \frac{\mathbf{x}_x - \mathbf{s}_x}{||\boldsymbol{\psi}_{xy}||^2} & 0 \end{bmatrix} \\ \nabla_{\mathbf{x}}\phi &= \begin{bmatrix} \frac{(\mathbf{x}_x - \mathbf{s}_x)(\mathbf{x}_z - \mathbf{s}_z)}{||\boldsymbol{\psi}||^2||\boldsymbol{\psi}_{xy}||} & \frac{(\mathbf{x}_y - \mathbf{s}_y)(\mathbf{x}_z - \mathbf{s}_z)}{||\boldsymbol{\psi}||^2||\boldsymbol{\psi}_{xy}||} & -\frac{||\boldsymbol{\psi}_{xy}||^2}{||\boldsymbol{\psi}||^2||\boldsymbol{\psi}_{xy}||} \end{bmatrix} \end{aligned} \tag{3.19}$$

The Jacobian described above may then be input into Equation (2.44) to produce an updated estimate of the subject's state as given by

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + (\mathbf{J}_{k-1}^\top \boldsymbol{\Sigma}^{-1} \mathbf{J}_{k-1})^{-1} \mathbf{J}_{k-1}^\top \boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1} \Delta\boldsymbol{\Omega} \tag{3.20}$$

This process may be iterated until it converges. The process is said to converge when the $\|\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k-1}\| < 1 \times 10^{-6}$. The residual term $\Delta\boldsymbol{\Omega}$ is the difference between the measured LOS and the LOS that should have been ideally measured if the subject's state is $\mathbf{x}_{k-1}$. This is expressed by

$$\Delta\boldsymbol{\Omega} = \boldsymbol{\Omega} - \boldsymbol{\Omega}\left(\mathbf{x}_{k-1}\right) \tag{3.21}$$

where $\boldsymbol{\Omega}$ is the measured LOS angles. This process requires an initial estimate to begin using the NLO at $k = 0$. The initial estimate is provided via triangulation as described in Section 2.3. The remaining unknown quantity from Equation (3.20) is the matrix $\Sigma_{\boldsymbol{\Omega}}$.

$\Sigma_{\boldsymbol{\Omega}}$ is the covariance matrix of the measurements. The covariance matrix is arbitrarily defined as

$$\Sigma_{\boldsymbol{\Omega}} = \sigma^2 I \tag{3.22}$$

where the standard deviation $\sigma = 5 \times 10^{-6}$ rad. (In Chapter 4 the change in geolocation performance resulting from difference standard deviations is discussed). With the covariance matrix defined, all of the terms in Equation (3.20) have been described.

The covariance matrix for the measurements may be combined with the Jacobian via Equation (2.52) to produce the covariance matrix of the subject's state as given by

$$\boldsymbol{\Sigma}_{\mathbf{x}} = \left(\mathbf{J}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{\Omega}}^{-1}\mathbf{J}\right)^{-1} \tag{3.23}$$

If the initial state estimate is close to the true state, this NLO will converge to an unbiased least-squares solution to the static geolocation problem. The next section will

address the construction of the Velocity NLO. The Velocity NLO optimizes for the subject's motion.

3.4 Velocity NLO

In this section, an NLO is developed to solve the geolocation problem where the subject's position changes significantly over a short amount of time. If all of the sensors measure the LOS to the subject at the same time and subject's state is estimated at that time, then the subject's velocity is irrelevant. In this case the geolocation problem simplifies to the static scenario. The more interesting problem is the scenario where the sensors do not measure LOS at the same time. In this case, a time at which to estimate the subject's state must be selected.

We let the time when a sensor produces a LOS measurement of the subject be t_m. The time at which the user desires to estimate the subject is t_e. For this type of LOS data, it is assumed that the subject's motion is approximately linear over short spans of time. Therefore, the subject's motion is described similar to Equation (3.10) by

$$\mathbf{p}(t) = \mathbf{p}(t)|_{t=0} + \dot{\mathbf{p}}t \tag{3.24}$$

where the subject's position is given by $\mathbf{p}(t)$ and its velocity is given by $\dot{\mathbf{p}}$. As before, the velocity is constant. There is now an additional unknown in this problem: the velocity of the subject. Therefore, the subject's state now includes velocity and is ordered as given by

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_x & \mathbf{x}_y & \mathbf{x}_z & \dot{\mathbf{x}}_x & \dot{\mathbf{x}}_y & \dot{\mathbf{x}}_z. \end{bmatrix}^\top \tag{3.25}$$

As before, the NLO requires that LOS measures are described in terms of the subject's state by a function herein called $\mathbf{\Omega}(\mathbf{x})$. In this case, the NLO is being used to estimate the subject's state at an estimation time which may or may not be the same as the measurement time. This relationship is developed in the manner illustrated in Figure 3.6. The subject's

position is projected in time via its model of motion to measurement time from the estimation time. The relationship between the measurements and the subject's state is then identical to that of the static case.

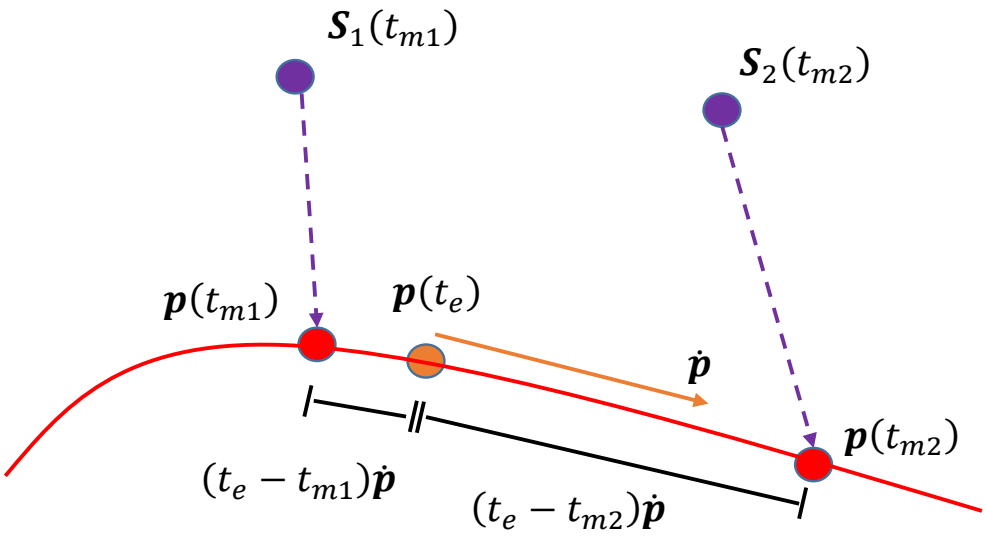

Figure 3.6: Setup for the velocity NLO. The subject's velocity is used to project its state from the estimation time to the measurement time.

From Figure 3.6, it can be seen that the difference between the subject's position at the measurement time and its position at the estimation time may be calculated as

$$\mathbf{p}(t_m) = \mathbf{p}(t_e) - (t_e - t_m)\dot{\mathbf{p}}. \tag{3.26}$$

Because the objective is to estimate the subject's state at t_e, the true LOS measurements associated with the subject's state at t_e must be calculated. If calculable, this allows the LOS measurements to be compared against the true LOS that should be measured if the

estimated subject state were its true state. The difference between the measured LOS and the true LOS associated with the estimate of the subject's state is the residual term $\Delta\boldsymbol{\Omega}$.

It is assumed that in a general scenario, the estimation time does not coincide with a measurement. Equation (3.26) gives an equation for the subject's state at the estimation time in terms of the measurement time. Therefore, the ideal measurements are a function of the subject's state at the measurement time in terms of its state at the estimation time. This is mathematically given by

$$\boldsymbol{\Omega}\left(\mathbf{x}(t_m)|\mathbf{x}(t_e)\right) = \boldsymbol{\Omega}\left(\begin{bmatrix} \mathbf{p}(t_e) - (t_e - t_m)\dot{\mathbf{p}} \\ \dot{\mathbf{p}} \end{bmatrix}\right) \tag{3.27}$$

As in the static case, the NLO's θ and ϕ measurement angles depend only on the subject's position at the measurement time. However, because the measurements are now described in terms of the subject's state (position and velocity in this case) at the estimation time, the subject's velocity must also be taken into account. Therefore, Equation (3.17) is modified for the scenario with a constant-velocity subject, ie

$$\begin{aligned} \boldsymbol{\psi} &= (\mathbf{p}(t_e) - \dot{\mathbf{p}}(t_e - t_m)) - \mathbf{s} \\ \theta &= \arctan 2\left(\boldsymbol{\psi}_y, \boldsymbol{\psi}_x\right) \\ \phi &= \arctan 2\left(\sqrt{\boldsymbol{\psi}_x^2 + \boldsymbol{\psi}_y^2}, \boldsymbol{\psi}_z\right) \end{aligned} \tag{3.28}$$

More than one LOS measurement is used for geolocation, so $\boldsymbol{\Omega}(\mathbf{x})$ is the concatenation of θ and ϕ and functions of the subject's state for each sensor. This concatenation of functions is linearly approximated using the Jacobian. As before, the Jacobian contains the first partial derivatives of θ and ϕ with respect to each variable in the subject's state for each sensor at the current estimate of the subject's state. In Equation (3.18) the Jacobian contained only partial derivatives with respect to the subject's position. Now the subject's

state includes its velocity, and so the Jacobian also contains the partial derivatives of θ and ϕ with respect to velocity. The Jacobian for the velocity NLO is given by

$$\mathbf{J}_{k-1} = \begin{bmatrix} \frac{\partial\theta_1}{\partial\mathbf{x}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_1}{\partial\mathbf{x}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_1}{\partial\mathbf{x}_z}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_1}{\partial\dot{\mathbf{x}}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_1}{\partial\dot{\mathbf{x}}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_1}{\partial\dot{\mathbf{x}}_z}\Big|_{\mathbf{x}_{k-1}} \\ \frac{\partial\phi_1}{\partial\mathbf{x}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_1}{\partial\mathbf{x}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_1}{\partial\mathbf{x}_z}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_1}{\partial\dot{\mathbf{x}}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_1}{\partial\dot{\mathbf{x}}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_1}{\partial\dot{\mathbf{x}}_z}\Big|_{\mathbf{x}_{k-1}} \\ \frac{\partial\theta_2}{\partial\mathbf{x}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_2}{\partial\mathbf{x}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_2}{\partial\mathbf{x}_z}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_2}{\partial\dot{\mathbf{x}}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_2}{\partial\dot{\mathbf{x}}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_2}{\partial\dot{\mathbf{x}}_z}\Big|_{\mathbf{x}_{k-1}} \\ \frac{\partial\phi_2}{\partial\mathbf{x}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_2}{\partial\mathbf{x}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_2}{\partial\mathbf{x}_z}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_2}{\partial\dot{\mathbf{x}}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_2}{\partial\dot{\mathbf{x}}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_2}{\partial\dot{\mathbf{x}}_z}\Big|_{\mathbf{x}_{k-1}} \\ & & \vdots & \vdots & & \\ \frac{\partial\theta_m}{\partial\mathbf{x}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_m}{\partial\mathbf{x}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_m}{\partial\mathbf{x}_z}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_m}{\partial\dot{\mathbf{x}}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_m}{\partial\dot{\mathbf{x}}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\theta_m}{\partial\dot{\mathbf{x}}_z}\Big|_{\mathbf{x}_{k-1}} \\ \frac{\partial\phi_m}{\partial\mathbf{x}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_m}{\partial\mathbf{x}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_m}{\partial\mathbf{x}_z}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_m}{\partial\dot{\mathbf{x}}_x}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_m}{\partial\dot{\mathbf{x}}_y}\Big|_{\mathbf{x}_{k-1}} & \frac{\partial\phi_m}{\partial\dot{\mathbf{x}}_z}\Big|_{\mathbf{x}_{k-1}} \end{bmatrix} \tag{3.29}$$

These partial derivatives are taken with respect to θ and ϕ as given in Equation (3.28). The result of this differentiation is similar to that given by Equation (3.19); however, the derivatives are too lengthy to be written in this document. The derivatives are recorded in VNLO_NTD.m given in Appendix A.

The covariance matrix of the measurements $\mathbf{\Sigma_\Omega}$ is defined just like in the static case. The NLO is then used to improve estimates of the subject's state according to Equation (3.20).

The NLO requires an initial estimate of the subject's state. Previously, the initial estimate contained only position. Now the initial estimate includes velocity. The initial guess of velocity is produced by interpolating between initial position estimates. The position estimates are provided via triangulation. The initial velocity estimate is calculated by

$$\hat{\dot{\mathbf{p}}}_0(t_{e_i}) = \frac{\hat{\mathbf{p}}_0(t_{e_i}) - \hat{\mathbf{p}}_0(t_{e_{i-1}})}{t_{e_i} - t_{e_{i-1}}}, \qquad i > 1 \tag{3.30}$$

where t_{e_i} is the i^{th} estimation time and the 0 subscript represents that this is the initial velocity estimate which is used to start the NLO. Note that because there is no estimation

prior to the first estimation time, the velocity at the first estimation time cannot be interpolated. Therefore, it is assumed that the velocity at the initial estimation time is the velocity at the subsequent estimation time.

Since the NLO also optimizes the subject's velocity, the covariance matrix of the subject's velocity is also produced. The covariance matrix of the subject's velocity may be found from the covariance matrix of the subject's state. The covariance matrix of the subject's state is given by Equation (2.52). This covariance matrix is expanded yielding

$$\mathbf{\Sigma_x} = \begin{bmatrix} \text{var}(\mathbf{p}_x) & \text{cov}(\mathbf{p}_y, \mathbf{p}_x) & \dots & \text{cov}(\dot{\mathbf{p}}_z, \mathbf{p}_x) \\ \text{cov}(\mathbf{p}_x, \mathbf{p}_y) & \text{var}(\mathbf{p}_y) & \dots & \text{cov}(\dot{\mathbf{p}}_z, \mathbf{p}_y) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(\mathbf{p}_x, \dot{\mathbf{p}}_z) & \dots & \dots & \text{var}(\dot{\mathbf{p}}_x) \end{bmatrix} \tag{3.31}$$

The upper left 3×3 matrix contains the position covariance matrix. The lower right 3×3 matrix contains the velocity covariance matrix. These covariance matrices may be used to produce an error ellipsoid for position and velocity via the procedure given in Section 2.5. In the next section, the subject is still modeled as having an approximately linear path over short amounts of time; however, the time delay is added to the model.

3.5 Time-Delay and Velocity NLO

This section addresses the construction of the time delay and velocity NLO. The previous NLO assumed that the subject's motion may be approximately modeled as having constant velocity over a short length of time. That model may be more realistic for fast-moving subjects. However, it ignores another factor in the relationship between the measurements and the subject: the time delay.

The sensors modeled in this research are passive, and therefore, they produce LOS measurements of the subject from the emission radiated by the subject. The propagation

time between the subject and the sensor is referred to as the time delay. The time delay is given the variable t_d throughout this thesis.

The first step in developing the NLO is to develop a description of a sensor's measurements in terms of the subject's state. Equation (3.14) is the equation used to simulate a time delay for a subject state and sensor position. In that case, data is simulated and the subject's path is known. The subject's state is unknown for the NLO; however, the equation still provides an equation for t_d in terms of the subject's state $\mathbf{x}$. The goal is to describe the measurements that a sensor should ideally receive in terms of the subject's state. This function is depicted as $\mathbf{\Omega}(\mathbf{x})$. Since the time delay is explicitly described in terms of the subject's state in Equation (3.14) it is not another element of the subject's state.

For the velocity NLO, the ideal measurement angles are a function only of the subject's position at the time of measurement t_m. When time delay is considered, the ideal measurements are rather only a function of the subject's position at the time when the subject emitted the radiation $(t_m - t_d)$ observed by the sensor. Furthermore, it may be desirable to geolocate the subject at a time other than a measurement time.

This issue is encountered in the previous section where the velocity NLO is developed. The time at which the subject's state is desired is called the estimation time t_e. For the velocity NLO, the subject's state at the time of measurement is described in terms of its state at the desired estimation time. The same approach is followed here the only difference being that the subject's location at the time of measurement is not the location that the sensor's LOS measures.

Figure 3.7 develops the solution to this problem in the same manner as for the velocity NLO. The objective is to develop the function $\mathbf{\Omega}(\mathbf{x}(t_e))$. $\mathbf{\Omega}(\mathbf{x}(t_e))$ describes the LOS measurements as a function of the subject's state at the estimation time. The sensor's measurements at t_m are measurements of the subject's location at t_m minus the time it took

for the radiation to reach the sensor from the subject t_d. Therefore, the measurements at the measurement time is the LOS to the subject at $t_m - t_d$. Therefore, the measurements angles θ and ϕ may be described as

$$\boldsymbol{\psi} = (\mathbf{p}(t_m - t_d) - \mathbf{s}(t_m)$$
$$\boldsymbol{\Omega}(\mathbf{x}(t_e)) = \begin{bmatrix} \theta(\mathbf{x}(t_e)) \\ \phi(\mathbf{x}(t_)) \end{bmatrix} = \begin{bmatrix} \arctan 2\left(\boldsymbol{\psi}_y, \boldsymbol{\psi}_x\right) \\ \arctan 2\left(\sqrt{\boldsymbol{\psi}_x^2 + \boldsymbol{\psi}_y^2}, \boldsymbol{\psi}_z\right) \end{bmatrix} \tag{3.32}$$

where $\mathbf{s}(t_m)$ is the sensor's position at t_m, $\mathbf{p}$ is the subject's location, t_d is the time delay, and $\boldsymbol{\psi}$ is the difference vector between the subject and the sensor.

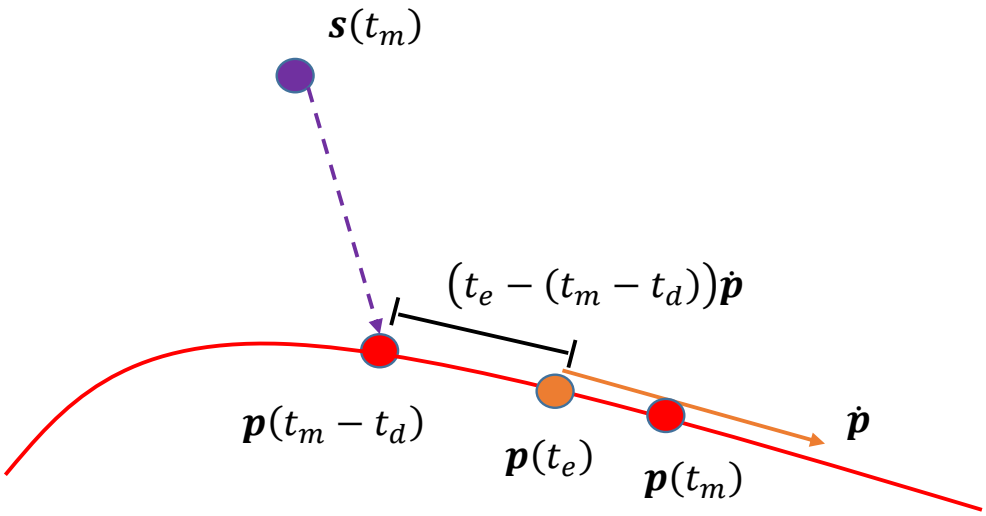

Figure 3.7: Geometry for time delay and velocity NLO development. The velocity is used to project the subject from its location at t_m to its location corresponding to the sensors' measurements. It is then projected to its position at the estimation time.

Equation (3.32) gives the noiseless LOS measurement that should be produced in terms of the subject's state at $t_e - t_d$. To estimate the subject's state at t_e via the NLO,

the ideal measurements must be described in terms of the subject's state at t_e. Therefore, a relationship between the state at $t_e - t_d$ and t_e must be found. It is assumed that over short durations, the subject's motion is approximately linear, and so it has constant velocity, ie

$$\mathbf{p}(t_m - t_d) = \mathbf{p}(t_e) - \dot{\mathbf{p}}(t_e - (t_m - t_d)) \tag{3.33}$$

This equation still contains an unknown, the time delay t_d. In Section 3.2.3, Equation (3.14) is developed in which an equation for the time delay is described in terms of the subject's state at t_m. However, to write $\mathbf{\Omega}(\cdot)$ as a function exclusively of $\mathbf{x}(t_e)$, t_d must be described in terms of $\mathbf{x}(t_e)$. The solution for t_d from Equation (3.14) is denoted as function $t_d(\mathbf{x}(t_m))$. Given the linear path assumption for the subject, we have

$$\mathbf{p}(t_m) = \mathbf{p}(t_e) + \dot{\mathbf{p}}(t_m - t_e). \tag{3.34}$$

Therefore, the time delay function may be described in terms of the subject's state at t_e as given by

$$t_d(\mathbf{x}(t_m)) = t_d(\mathbf{p}(t_e) + \dot{\mathbf{p}}(t_m - t_e), \dot{\mathbf{p}}) \tag{3.35}$$

For simplicity this result will be called $t_d(\mathbf{x}(t_e))$.

When Equation (3.35) is coupled with Equation (3.32) and Equation (3.33) it produces the following equation for the ideal measurements exclusively in terms of the subject's state at t_e as given by

$$\begin{aligned} \boldsymbol{\psi} &= \left[\mathbf{p}(t_e) - \dot{\mathbf{p}}\left(t_e - t_m + t_d(\mathbf{x}(t_e))\right)\right] - \mathbf{s}(t_m) \\ \mathbf{\Omega}\left(\mathbf{x}(t_m)\right) &= \begin{bmatrix} \theta(\mathbf{x}(t_m)) \\ \phi(\mathbf{x}(t_m)) \end{bmatrix} = \begin{bmatrix} \arctan 2\left(\boldsymbol{\psi}_y, \boldsymbol{\psi}_x\right) \\ \arctan 2\left(\sqrt{\boldsymbol{\psi}_x^2 + \boldsymbol{\psi}_y^2}, \boldsymbol{\psi}_z\right) \end{bmatrix} \end{aligned} \tag{3.36}$$

The linearity assumption in the NLO results from finding the Jacobian of this equation. Because the time delay was described in terms of the subject's state at the t_e, the time delay is not another element of the state vector. Therefore, the Jacobian is formed just as given by Equation (3.29) for the velocity NLO. However, the differentiation of θ and ϕ in terms of the subject's state is not the same as it was for the Velocity NLO because the time delay is included.

Given this Jacobian, the NLO iteratively improves upon its initial guess via Equation (3.20). The only difference is in the calculation of the Jacobian. Furthermore, the addition of the time delay does not change the procedure for producing the error surface or subject's state covariance matrix. One issue that has not been discussed thus far is the selection of measurements which are used to estimate the subject's state.

3.6 Measurement Selection

The NLO linearizes an inherently nonlinear problem. The linearization transforms the problem into iteratively solving an overdetermined system of equations as described in Section 2.4. Ignoring weighting, the NLO is set up as

$$\Delta\mathbf{\Omega}_k \approx J_k \Delta\mathbf{x}_k \tag{3.37}$$

where J is the Jacobian which provides the linearizing assumption as described in the previous sections. $\Delta\mathbf{x}_k$ is the difference between the current best estimate of the subject's state $\mathbf{x}_k$ and the next best estimate. $\Delta\mathbf{\Omega}$ is the difference between the measured angles $\mathbf{\Omega}$

and the LOS angles associated with $\mathbf{x}_k$. Equation (3.37) is expanded yielding

$$\Delta\boldsymbol{\Omega}_k = \begin{bmatrix} \theta_1 \\ \phi_1 \\ \theta_2 \\ \phi_2 \\ \vdots \\ \theta_m \\ \phi_m \end{bmatrix} - \begin{bmatrix} \theta_1(\mathbf{x}_k) \\ \phi_1(\mathbf{x}_k) \\ \theta_2(\mathbf{x}_k) \\ \phi_2(\mathbf{x}_k) \\ \vdots \\ \theta_m(\mathbf{x}_k) \\ \phi_m(\mathbf{x}_k) \end{bmatrix} \tag{3.38}$$

where θ_j is the θ angle that was measured by the j^{th} sensor and $\theta_j(\mathbf{x}_k)$ is the angle that would have been measured in a noiseless scenario if the subject's state really was $\mathbf{x}_k$.

For the system to be overdetermined, the dimension of $\boldsymbol{\Omega}$ must be greater than the number of elements in the subject's state. Therefore, if there are n elements in the subject's state, then $2 \times m > n$. Therefore, the static NLO requires that a minimum of two measurements be used. The velocity NLO and time delay NLO requires three measurements to be used. However, more measurements could be used.

There may be many measurements taken over a long timespan. All of these measurements could potentially be used, too. For many types of subjects, this produces large errors because of the assumptions inherent to the various NLOs.

The static NLO assumes that the subject is approximately stationary over the length of time that the measurements used by the NLO are produced. This assumption may be terribly inaccurate over a long period of time, so the resulting estimate would be very inaccurate. If the subject's flightpath is not approximately linear over the entire duration that it is observed, the estimate produced by the velocity NLOs would be inaccurate. Therefore, only a subset of the measurements should be used to estimate the subject.

The process of selecting the measurements that should be used to geolocate the subject is called windowing. This is performed via the function "setWindow.m" provided in Appendix A. There are several factors to consider when deciding which measurements should be used to geolocate the subject at a certain time.

Suppose one sensor produces several LOS measurements in a short time. Because the sensor's location has not changed significantly, these LOS measurements are not very different from one another, and so they may produce an inaccurate geolocation estimate. In terms of the confidence surface, these measurements will be incapable of narrowing the error surface in range. Such a scenario may provide a good estimate in two dimensions, but its estimate in range may be very inaccurate. This fact is illustrated in Figure 3.8.

Therefore, it is desirable to use multiple sensors with different views of the subject. A diversity in observation directions provides information on the subject's position in all dimensions. Such measurements are referred to as diverse.

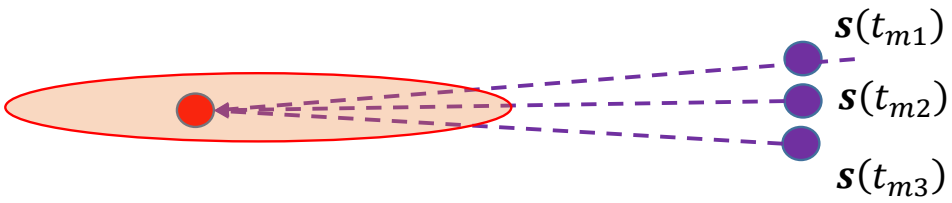

Figure 3.8: Effect of using NLO with several similar LOS measurements. Since there is very little diversity in the measurements in range, the range estimate may be inaccurate. For this reason, it is desirable to use multiple independent sensors with different look-angles.

While unlikely, a situation with limited viewing diversity is possible with multiple sensors. This possible scenario is ignored in the windowing process. If multiple sensors are used to provide diversity, then multiple measurements from the same sensor might also

be used. However, measurements that weren't taken near the estimation time may cause inaccurate geolocation since the NLOs assumptions become inaccurate.

Therefore, three criteria are used in performing the windowing. The window must provide a minimum number of measurements, $n_{\mathbf{\Omega}}$, some of which may be from the same sensor. The measurements must come from a minimum number of unique sensors $n_{\mathbf{s}}$, and a maximum window size t_w. When estimating the subject's state at a time t_e, the window selects measurements as described follows.

The window looks within $t_w/2$ of the estimation time. The window then selects the measurements that are closest to the estimation time that are taken from $n_{\mathbf{s}}$ different sensors such that there are a total of $n_{\mathbf{\Omega}}$ measurements.

For example, consider the following situation. The subject is being estimated at 0 seconds, and the requirements are a minimum of three measurements and two measurements from unique sensors. Suppose sensor one produces four measurements at 0, 1, 2 seconds and then sensor two produces a measurement at 5 seconds. If the window looks within 5 seconds of the estimation time, all four of the measurements will be used to satisfy the unique sensors requirement. However, if the window requirement was 4 seconds, only the first sensor's measurements would be used, and a warning message would be produced. More examples are illustrated in Figure 3.9 below.

The x-axis represents time. The red dots represent that times at which the subject's state is estimated. The other dots represent the sensors which produced measurements at the times indicated on the x-axis. The window for selecting the measurements to estimate the subject at each point in time are shown as the red-boxes. The parameters for the windowing are shown at the top of the figure.

Due to the timing of the LOS generation detailed in Section 3.2, the measurement constraints could also be satisfied without violating the window requirement. This simplification is done to reduce the number of random variables affecting the comparisons

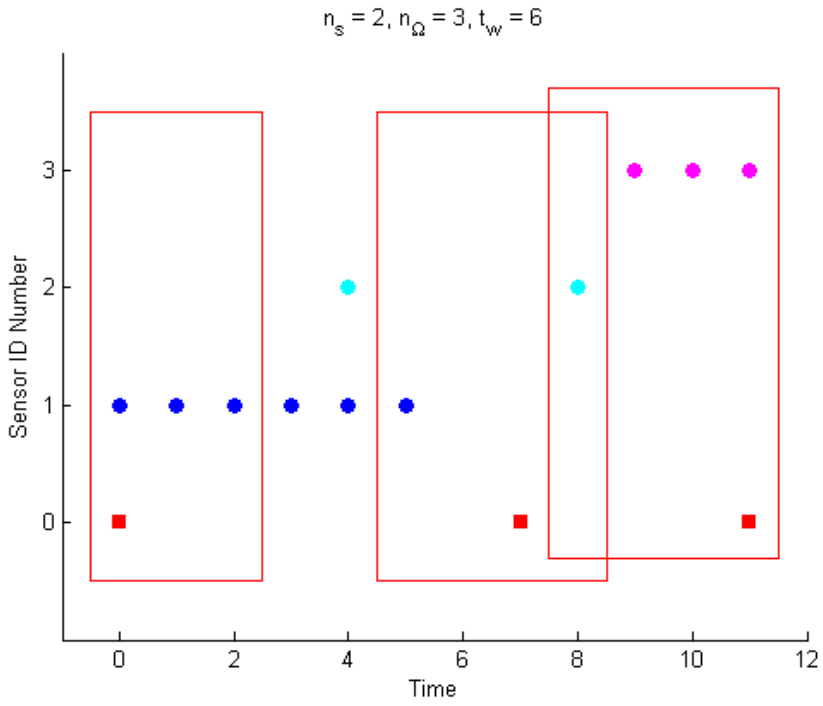

Figure 3.9: Three windowing examples. The x-axis is the temporal axis. The red squares on the lowest line represent estimation times. Each row represents measurement times for a unique sensor designed by a sensor ID number. The window for each estimation time is shown as a red box.

between the NLO algorithms. The following section describes how each algorithm's performance is objectively compared.

3.7 Algorithm Performance Metrics

3.7.1 Bias.

Metrics for describing the geolocation performance of the algorithms need to be defined. As discussed in section 2.4, the NLOs estimates should be unbiased if their model is accurate. The bias is in the algorithms is approximated via MCS. The average estimate of the subject's position or velocity is compared to the subject's true position or velocity.

Therefore, the bias β is given by

$$\beta = ||E(\hat{\mathbf{x}}) - \mathbf{x}|| \tag{3.39}$$

where $\hat{\mathbf{x}}$ is an estimate of the subject's true state $\mathbf{x}$. The next section provides a metric that describes the average error in an estimate.

3.7.2 Mean Absolute Error.

The *absolute error* of a geolocation estimate is given by $||\mathbf{x} - \hat{\mathbf{x}}||$ where $\mathbf{x}$ is the true subject location, and $\hat{\mathbf{x}}$ is an estimate of $\mathbf{x}$. This metric provides the absolute error of a measurement only for a specific system and a specific set of measurements. The *system* in this context means the locations of the sensors, their measurements and the subject's state. Because the measurements are random, the absolute error for a specific system is also random. Therefore, a useful metric for the estimate error of the algorithm needs to describe the absolute error of the algorithm for an arbitrary system and the random measurements.

While the distribution of the measurements is assumed to be Gaussian as per Chapter 2, the covariance of that distribution is unknown. To remove extra stochastic parameters from the algorithm's overall absolute error, the algorithm's absolute error is given for a fixed measurement covariance. Therefore, the algorithm's absolute error is described by the Mean Absolute Error (MAE) for a fixed covariance and a random system as given by

$$\xi = E\left[||\mathbf{x} - \hat{\mathbf{x}}||\right] \tag{3.40}$$

where ξ is the algorithm's MAE. The precision of a geolocation algorithm will be defined similarly.

3.7.3 Precision.

The precision of a specific estimate is analytically described by the covariance matrix $\mathbf{\Sigma}_{\mathbf{x}}$. It may be difficult to compare the precision entailed by a covariance matrix by the

entries of the matrix. Therefore, it is desirable to convert the covariance matrix into a single number.

A covariance matrix describes the shape of an ellipsoid [14]. Therefore, the precision may be described the volume of the ellipsoid. The volume of an ellipsoid is

$$V = \frac{4}{3}\pi abc \tag{3.41}$$

where a, b, and c are the lengths of the principal axes of the ellipsoid [21]. The principal axes of an ellipsoid are given by the eigenvectors and eigenvalues of the covariance matrix [18]. The eigenvectors are the principles axes, and their lengths are given by the square root of the eigenvalues associated with these eigenvectors. (Because covariance matrices are positive definite, their eigenvalues are always positive.) The eigenvalues and eigenvectors may be found via Singular Value Decomposition (SVD) [19].

The SVD decomposes the covariance matrix $\mathbf{\Sigma}$ into three parts: $\mathbf{U}$, $\mathbf{S}$, and $\mathbf{W}$. For a symmetric matrix such as a covariance matrix, $\mathbf{U}$ and $\mathbf{W}$ are identical [31]. The columns of $\mathbf{W}$ are the eigenvectors of $\mathbf{\Sigma}$. $\mathbf{S}$ is a diagonal matrix of the singular values of $\mathbf{\Sigma}$. The singular values of a matrix $\mathbf{\Sigma}$ are the square roots of the eigenvalues of $\mathbf{\Sigma}^H\mathbf{\Sigma}$ [25]. The objective is to associate the singular values of $\mathbf{\Sigma}$ with the eigenvalues of $\mathbf{\Sigma}$. Because covariance matrices are Hermitian $\mathbf{\Sigma}^H\mathbf{\Sigma} = \mathbf{\Sigma}^2$. Furthermore, if (λ,v) is an eigenpair of $\mathbf{\Sigma}$ associated with the eigenvector v then

$$\begin{aligned} \mathbf{\Sigma}\mathbf{v} &= \lambda\mathbf{v} \\ \mathbf{\Sigma}^2\mathbf{v} &= \mathbf{\Sigma}\mathbf{v}\lambda \\ \mathbf{\Sigma}^2\mathbf{v} &= \lambda^2\mathbf{v}. \end{aligned} \tag{3.42}$$

Therefore, if a matrix $\mathbf{\Sigma}$ has an eigenpair $(\lambda, \mathbf{v})$, then $\mathbf{\Sigma}^2$ has the eigenpair $(\lambda^2, \mathbf{v})$. The singular value σ of $\mathbf{\Sigma}$ associated with $\mathbf{v}$ is then given as

$$\sigma = \sqrt{\lambda\lambda^*} = |\lambda|. \tag{3.43}$$

From [18], the lengths of the ellipsoid associated with the covariance matrix are the square roots of the eigenvalues of the covariance matrix. Therefore, the lengths of the ellipsoid's principle axes are given by the square roots of the singular values of the covariance matrix.

Because the volume metric is variable with the system and the LOS measurements, an overall precision metric is defined as the expected value of the confidence surface's volume for a random system and a set measurement covariance matrix. Therefore, the precision metric is given as

$$\rho = E\left[V\right]. \tag{3.44}$$

While the precision metric provides a means of measuring the precision of a geolocation algorithm, this information has limited utility if the covariance matrix is not accurate.

3.7.4 Confidence Estimation Accuracy.

A geolocation algorithm may overestimate or underestimate the confidence in its estimate of the subject's state. Therefore, it is important to know whether or not an algorithm accurately represents its confidence estimate. The accuracy of a geolocation algorithm's confidence estimates may be calculated via the unitless Normalized (state) Estimation Error Squared NEES. The NEES is given in [1]. The NEES for a given measurement is defined as

$$\epsilon = (X - \hat{X})^\top \mathbf{\Sigma}_\mathbf{x} (X - \hat{X}). \tag{3.45}$$

If the geolocation algorithm accurately estimates the confidence in the subject's state (the covariance matrix), then $E[\epsilon]$ will equal the degrees of freedom in the subject's state. A larger value indicates that the algorithm produces overly confident estimates. Smaller values mean that the algorithm underestimates its confidence in the subject's state.

This chapter described the methodology by which the research question is evaluated. It described the simulations and models used to synthesize data for the geolocation algorithms. The next chapter describes the performances of the geolocation algorithms using synthesized data.

IV. Data & Results

This chapter provides the data and results for each of the algorithms developed in the previous chapter. These algorithms are the static NLO, velocity NLO, and time delay and velocity NLO. Each algorithm assumes certain characteristics for the subject. For example, the static NLO assumes that the subject is stationary. Furthermore, several claims were made regarding NLO performance when its assumptions are accurate. As described in Chapter 2, NLO should produce an unbiased geolocation estimate that is the most likely estimate, and NLO should accurately represent the confidence in its geolocation estimates.

Therefore, subsequent sections provide results with the objective of verifying that the NLOs converge to this estimate when their assumptions exactly model the scenario. Scenarios where these assumptions are true are simulated using the models described in Chapter 3. The setup for these scenarios is discussed and the results gathered from these scenarios are provided.

After evaluating the NLOs where their assumptions are accurate, the NLOs are used to geolocate the most challenging subject: the Space Vehicle (SV). The NLO's assumptions are only approximately true for this subject. Results on the relative performance of these algorithms on this subject are given. The results of this section establish a means of determining if the more robust NLOs provide improved geolocation accuracy, mean absolute error and precision. The first section of this chapter discusses the verification results for the static NLO.

4.1 Static NLO

The static NLO assumes that the subject is stationary. Therefore, simulated scenarios where the subject is stationary are used to verify the theoretical performance of the static

NLO. The generated scenarios consist of a random number of satellites. The satellites' positions are randomly generated such that their LOS is not occluded by the Earth. Each sensor is given a random orbiting radius. The LOS are then synthesized as discussed in Section 3.2. A typical system generated in this manner is shown in Figure 4.1.

Figure 4.1: Typical static NLO simulation. The magenta points are satellites. The red asterisk is the subject.

Figures 4.2 and 4.3 provide a visual comparison of the error surfaces provided by the NLO and TA. These error surfaces were generated as discussed in Sections 2.5 and 2.6 for one simulated scenario. These surfaces are shown for three standard deviations of error. Thus there was a 97% probability that the subject is located inside of the error surfaces.

The top portion of Figure 4.2 shows the size of the error surface for the triangulation method. The Monte Carlo error surfaces were generated using a MCS with 10,000 sample points. Compare the TA's error surface in the bottom plot to the top plot. The top plot shows the error surface for the NLO. Observe that the NLO's error surface is much smaller than the TA's error surface. The smaller size of the NLO's error surface means that the NLO provides tighter bounds (better confidence) in its estimate of the subject's position.

Furthermore, the top portion of Figure 4.2 illustrates another claim about the NLO: that the calculated error surface matches its true error. In Section 2.5, a method was

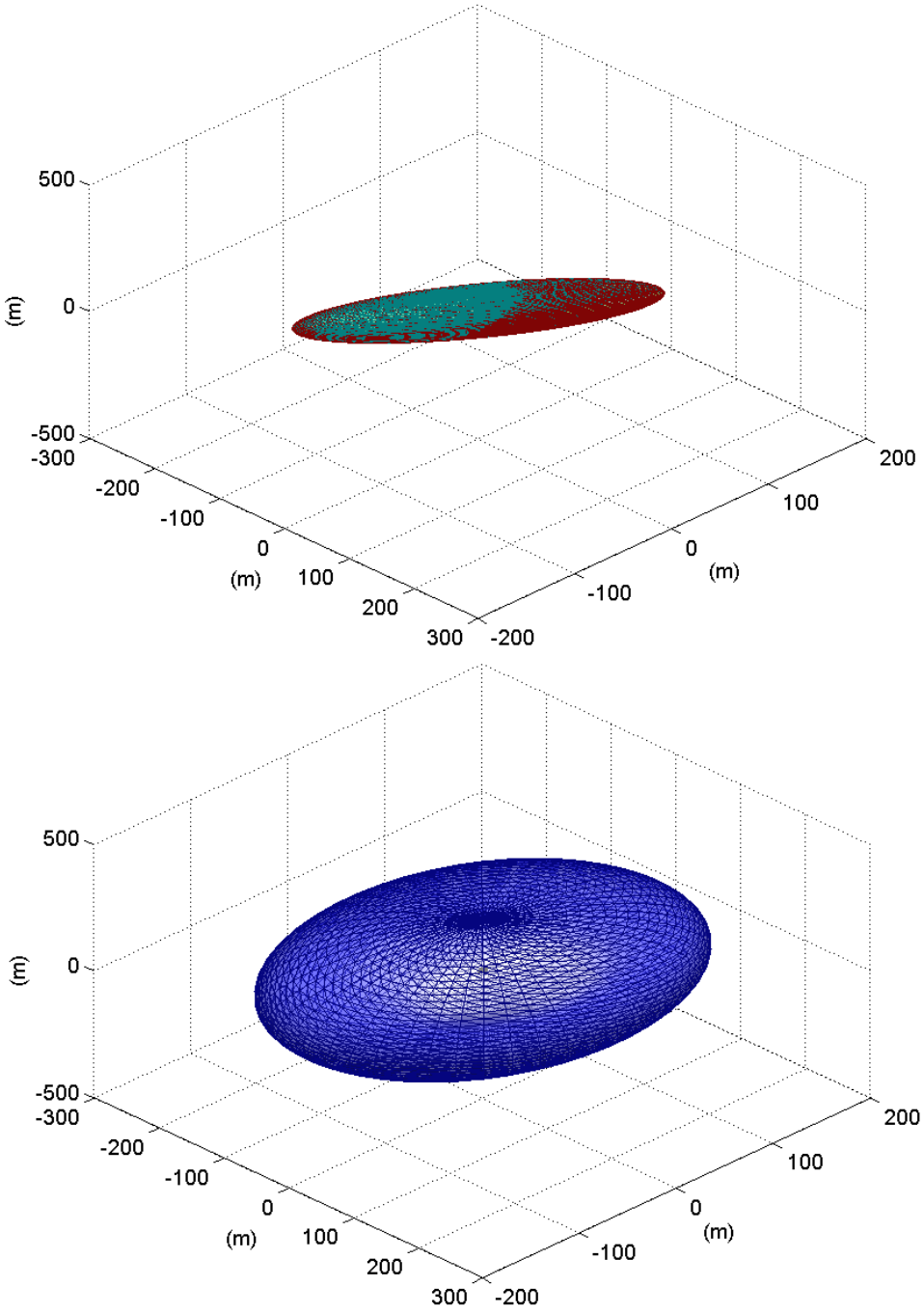

Figure 4.2: Error Surfaces for the Static NLO and the TA. The NLO's surface (top) is smaller (so the NLO provides better confidence). The NLO's calculated error surface (shown in red matches) its error surface produced using MCS (shown in teal).

provided for producing the NLO's error surface without using MCS. This method allowed the covariance matrix of the subject's state to be calculated rather than estimated by Monte Carlo Simulations. In this figure, the red surface is the NLO's calculated error surface and the teal error surface is produced by a MCS. It is visually clear that the calculated error surface and Monte Carlo error surface are nearly the same. This similarity indicates that the NLO accurately estimates its confidence. A comparison of the relative sizes of the error surfaces is shown in Figure 4.3.

In Figure 4.3, the TA and NLO's error surfaces are centered on their estimated subject position. The true subject position is shown in white. The NLO's error surface is red. The TA's error surface is blue. It is clear from this figure that the NLO's error surface is much smaller than that of the TA. From Section 3.7, the size of the error surfaces corresponds to the precision of the algorithms. Therefore, the NLO provides better precision (or confidence) in this scenario. The NLO is also centered much closer to the true subject position than the TA. Thus, the NLO is also more accurate in this scenario.

This figure provided a visual illustration of the performance error surface for the NLO and TA in a typical static scenario. Seven thousand of these static scenarios were simulated to produce statistics on the performance of the static NLO and the TA. Between 3 and 7 satellites were simulated in the scenarios. The performance of the TA and NLO will be estimated from these simulations. In Section 3.7, four statistics were discussed: bias, Mean Absolute Error (MAE), precision, and the confidence estimation accuracy.

The MAE is the average distance between the estimates of the subject's position and its true position. The precision is described as the average volume of the ellipsoid associated with some standard deviation of error. These values are given as a function of the number of sensors is given in Figure 4.4. It may not be easy to relate the volume of the ellipse to its size. Therefore, to provide better intuition into the ellipsoid sizes associated with these volumes, this figure also provides the length of the semi-major axis.

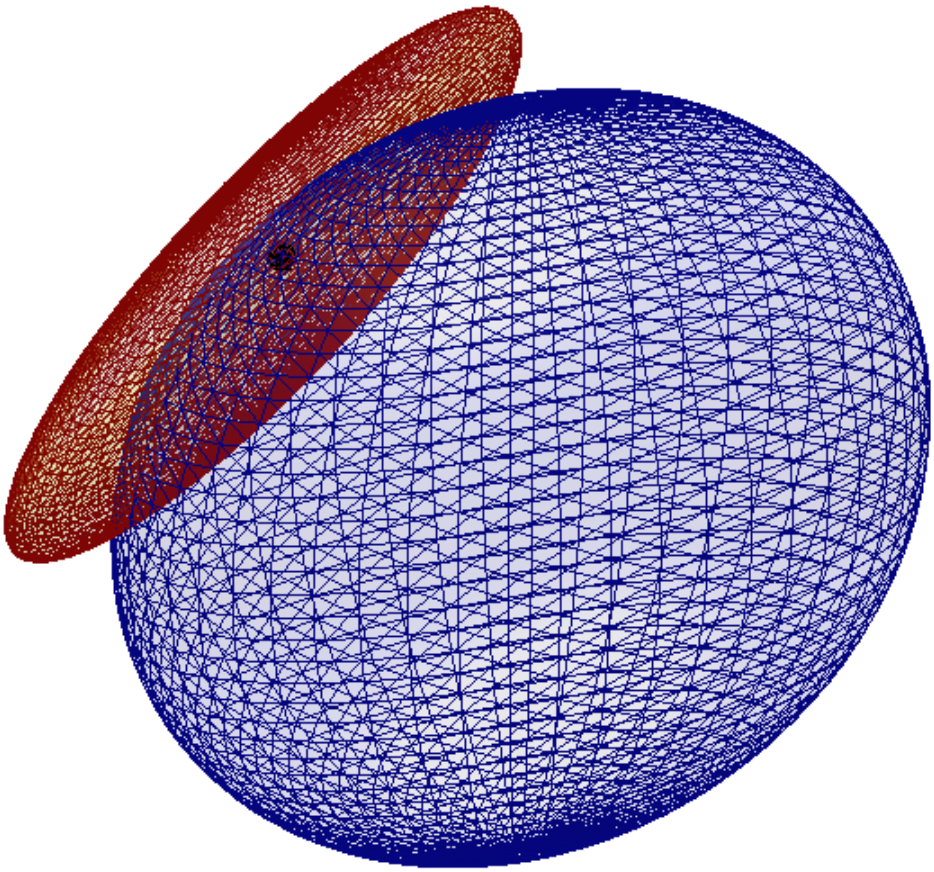

Figure 4.3: Combined static NLO and TA error surfaces. The true subject location is shown in black. The static NLO's error surface is shown in red. The TA's error surface is largest and is shown in blue.

The accuracy of the error surface estimates still needs to be discussed. The estimated error surface accuracy is described by the NEES. This metric does not depend on the number of sensors in the system, so it is not shown as a function of the sensor count. The means of the NEES for each of these algorithms are given by

	TA MCS	NLO MCS	NLO
μ_ϵ	3.04	3.04	3.02

As discussed in Section 3.7, if the mean of the NEES μ_ϵ converges to the degrees of freedom in the subject state, then the algorithm accurately estimates its error in the subject's state.

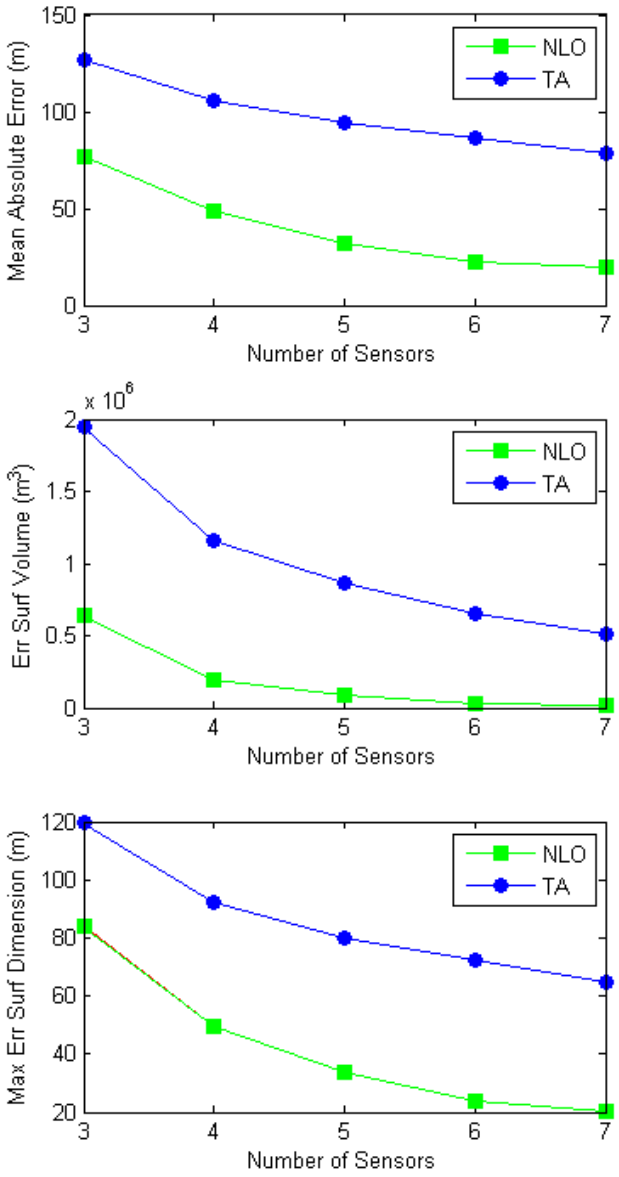

Figure 4.4: Static NLO and TA MAE and precision. The TA, NLO, and MCS of the NLO are shown in these plots. The MCS of the NLO tracks the NLO so closely that it cannot be seen in these plots.

In this case, the subject has three degrees of freedom: its x, y, and z coordinates. Thus, the MCS accurately estimates the TA's error surface. The NLO MCS accurately estimates its error surface, and the error surface calculated by the NLO matches its MCS. Thus, the NLO accurately calculates its error surface in these static scenarios.

In Section 2.4, it was claimed that the NLO produces unbiased estimates of the subject's state. Thus, the NLO's average estimate of the subject's state should equal the subject's true state. This is addressed in Figure 4.5.

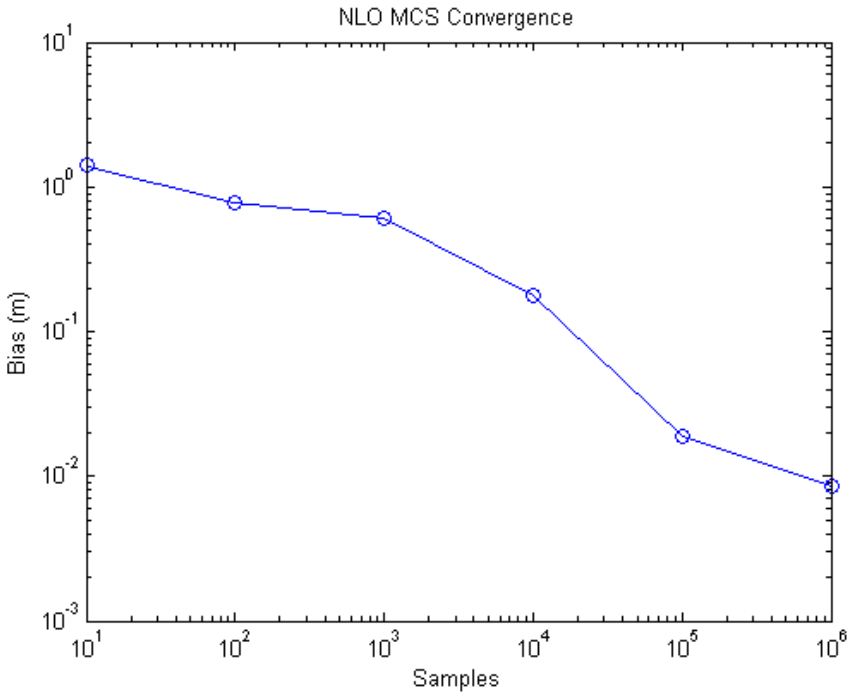

Figure 4.5: NLO is an unbiased estimator. LOS measurements about the true LOS are generated via MCS. The average of the resulting estimates of the subject's location is compared to the true location. The MCS is performed for different numbers of samples.

This figure shows the convergence of MCS for the NLO. Noisy LOS measurements centered on the true LOS were simulated. The NLO was used with these measurements. The average of the NLO's estimate of the subject's position was calculated as a function of

the number of Monte Carlo Samples. This figure illustrates that as more samples are used, the average NLO estimate approaches the subject's true position.

This section provides and discusses the results for the static NLO algorithm. The algorithm was tested in many scenarios where all of the Static NLO's assumptions are true: where the subject is stationary. Under these conditions the simulated performance should match the theoretical performance. A similar discussion is provided for the velocity NLO in the next section.

4.2 Velocity NLO

As in the previous section, the objective of this section is to demonstrate that the simulated performance of the velocity NLO matches the theoretical NLO. The velocity NLO is based on the assumption that the subject's motion is approximately linear over short durations of time. Furthermore, sensor measurements are not necessarily simultaneous. These scenarios are simulated as described in Section 3.4.

A subject is simulated with a linear flightpath and a constant velocity. Realistic sensors and their noisy measurements are also simulated. The velocity NLO is used with this data, and statistics are produced.

As described in Section 3.6, when sensor measurements are not simultaneous, a time at which to estimate the subject's state must be chosen. Furthermore, selecting measurements from which to estimate the subject's state are selecting via the measurement window. The measurement window has three settings. The settings used to produce the data for the Velocity NLO are given below.

A total of five satellites were generated. A minimum of seven measurements were used to perform the NLO. These seven measurements must come from at least three independent satellites, and the time window was set at 20 seconds. In addition to using the velocity NLO, the static NLO and TA were also used to geolocate the constant velocity subject.

The static NLO and TA do not account for velocity, so the MAE of their geolocation estimates should degrade as the subject's velocity increases. Because only velocity has been added to the system model, these statistics illustrate the added MAE due to velocity in the static NLO and TA.

To observe these effects as a function of the subject velocity, the subject's velocity was sampled. A constant velocity subject was simulated with magnitudes of 10, 50, 100, 500, 1000, 3000, and 7000 meters per second. At each velocity, twenty scenarios were produced from which more than 10,000 state estimates were recorded to generate statistics.

Figure 4.6 shows the MAE in position estimates for the three geolocation algorithms versus subject velocity. Note that the constant velocity NLO's estimates maintain the same MAE irregardless of increased subject velocity. This is expected since the NLO optimizes over velocity. Further observe that the static NLO provides the smallest MAE for very slow subjects; however, its MAE degrades the fastest. Once the subject's velocity is greater than 1000 meters per second, the static NLO appears to diverge. This means that the static NLO's assumption that the subject is approximately stationary is so inaccurate that it no longer converges. The TA's estimates are more accurate than the static NLO at faster velocities, but it performs worse than the velocity NLO.

The MAE of the velocity estimates is similar to the error for the position estimates. This is shown in Figure 4.7. As expected, the velocity NLO provides constant MAE in its velocity estimates. Observe the same divergent behavior in static NLO, and the TA's decreased MAE. Note that neither the static NLO nor the TA innately calculate velocity. Their velocity estimates were calculated by interpolation. A remaining question is the mean accuracy of the error surface estimates for the NLOs.

Figure 4.8 shows the NEES for the static NLO as a function of subject velocity. The NEES starts at 3.1 (slightly over confident) and ends with a value near 10^8. For such large velocities, the static NLO is drastically over confident. The NEES of the velocity NLO was

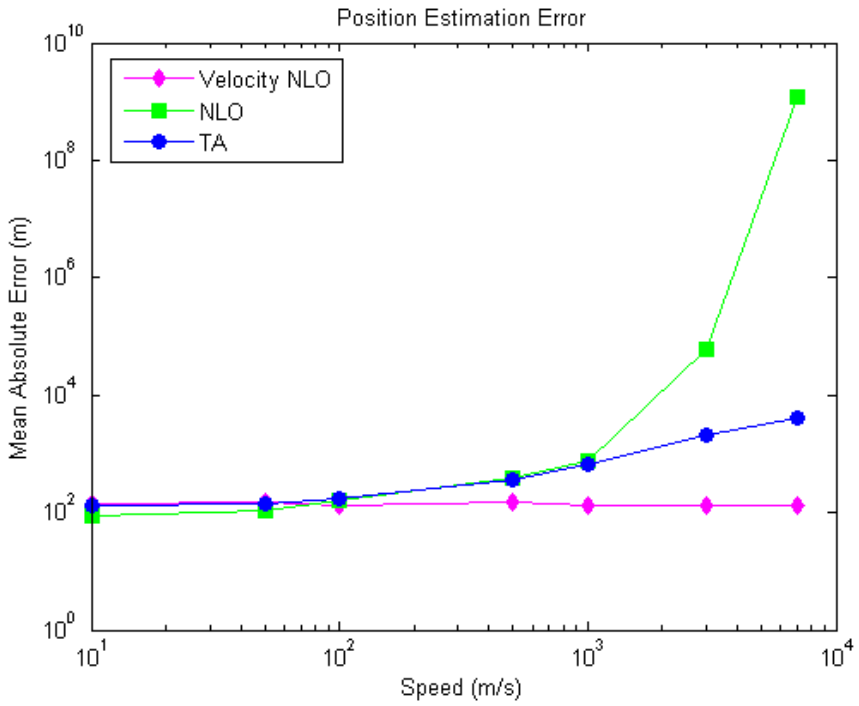

Figure 4.6: Mean absolute error in the estimated position for a constant velocity subject without time delay. The velocity NLO has a constant mean absolute error as expected. The TA and the static NLO's mean absolute error degrades with increased velocity.

estimated to be six. Note that since the velocity NLO optimizes over position and velocity, it estimates six parameters of the subject's state. Therefore, an NEES of six is ideal. The velocity NLO accurately calculates its error surface for all subject velocities.

The velocity NLO should also be an unbiased estimator meaning that its geolocation estimates should be unbiased. Furthermore, because the static NLO does not include subject motion in its model, its geolocation estimates may be biased. Therefore, MCS were used where the LOS measurements of the subject were simulated about the true LOS. The static and velocity NLOs were used with this data to geolocate the subject. The result of this MCS is shown in Figure 4.9.

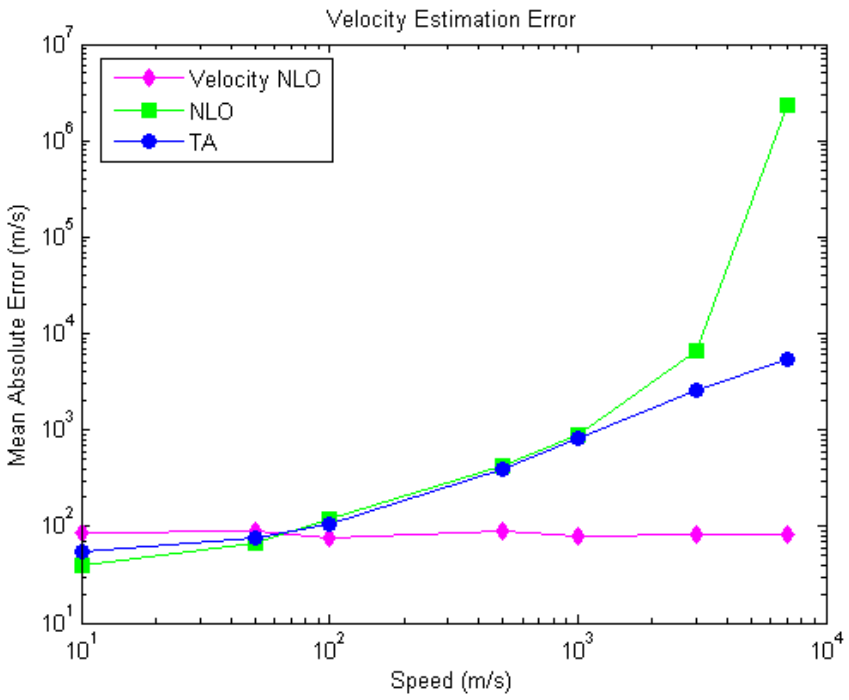

Figure 4.7: Mean absolute error in velocity estimates for the velocity NLO, static NLO, and TA. The velocity NLO performs as expected. The error in the other algorithms' geolocation estimates increases with the subject's velocity.

The results provided in this section describe the performance of the TA and static and velocity NLOs. The velocity NLO's simulated performance matches its theoretical performance. The next section describes a similar comparison among the algorithms when the time delay is included.

4.3 Time-Delay and Velocity NLO

This section examines the performance of the time delay and velocity NLO. For brevity in this section, the time delay and velocity NLO will be referred to as NLO_{TD}. As in the previous sections, the primary objective of this section is to determine if the simulated performance of the NLO_{TD} matches its theoretical performance. Furthermore, the previous algorithms do not account for time delay, so this section illustrates the added error in the previous algorithms due to the time delay.

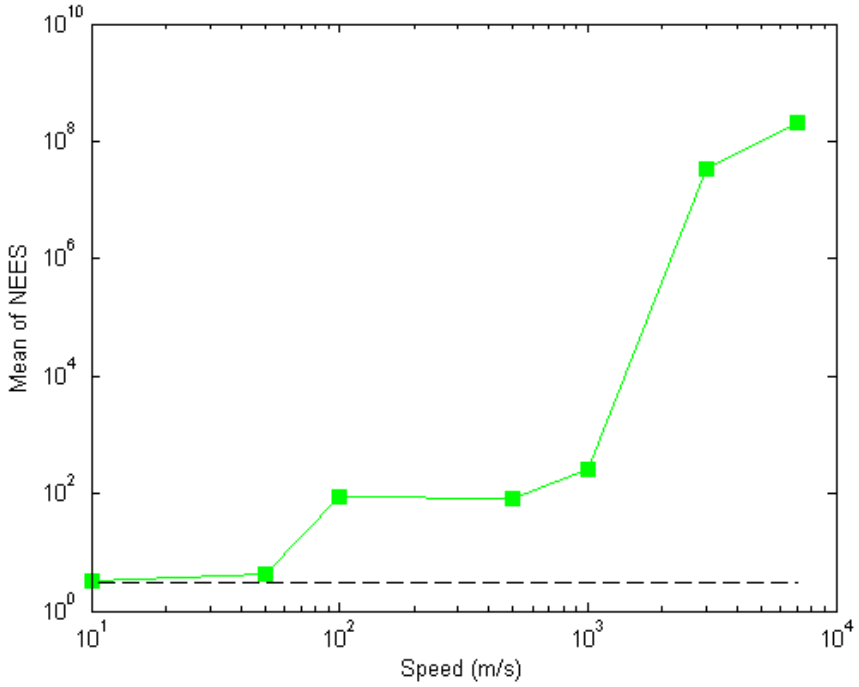

Figure 4.8: The static NLO's NEES for the constant velocity subject without time delay. The NEES grows as the subject's speed increases. This means that the static NLO drastically over estimates its confidence in its estimates.

The simulated subject in this section is identical to the linear subject in the previous section, except that the time delay affects the LOS measurements. These scenarios consist of a single subject moving with constant velocity. It is observed by five sensors. To estimate the subject, a minimum of seven measurements are used from at least three unique sensors. All of these measurements must be found within 20 seconds of the estimation time.

If unaccounted for, the time delay creates a bias in the LOS measurements. This bias increases with the subject's velocity. Bias is induced into the measurements because the subject moves some distance during the time that it takes for its emission to be observed by the sensors. The faster the subject moves, the further it has moved over this time. Therefore, a LOS measurement produced at a certain time will not correspond to a measurement of the subject at that time. Thus the LOS measurements lag the position of the subject creating the

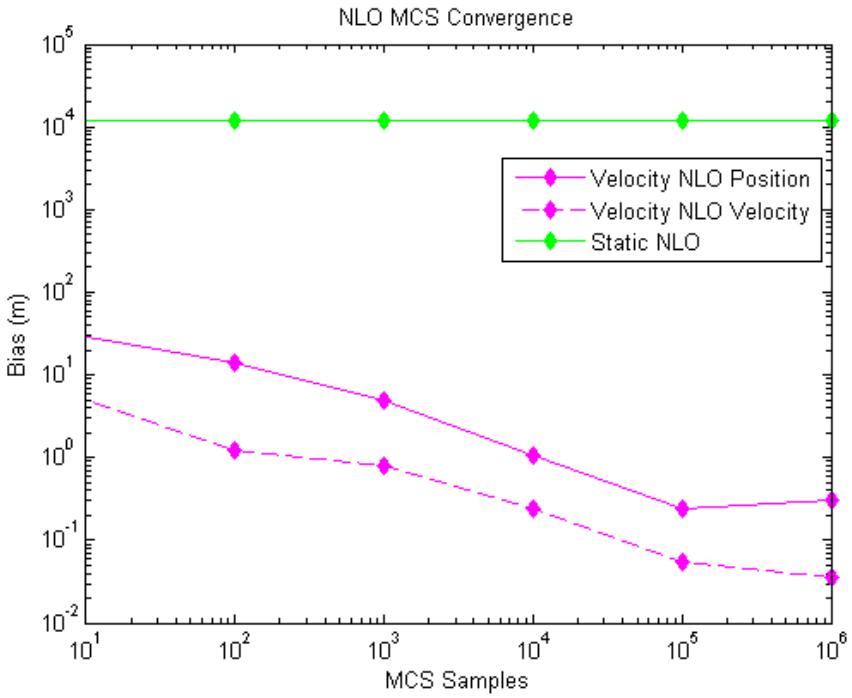

Figure 4.9: The velocity NLO is an unbiased estimator for a linear subject without time delay.

bias. The following result elucidates the magnitude of this error in estimating the subject's location as a function of velocity.

Subjects were simulated with velocities of 10, 50, 100, 500, 1,000, 3,000, and 7,000 meters per second. Statistics at each of these velocities are based on more than 10,000 geolocation samples. The TA, static and velocity NLOs, and NLO_{TD} were used to estimate the subject.

Figure 4.10 shows the MAE of the position estimates for each geolocation algorithm as a function of the subject's velocity. TheMAE of the NLO_{TD} should be unaffected by velocity because it accounts for the time delay. The other three algorithms do not, and so their performance should degrade as the subject's velocity increases. Once again, the static NLO performs the best for slower subjects; however, its performance degrades quickly as the subject's speed increases. The velocity NLO performs nearly as well as the NLO_{TD}

for all but the fastest subjects. For the fastest subject, the NLO_{TD} provides an MAE of 147 meters versus 654 meters for the velocity NLO and 4,691 meters for the TA. Similar results are shown for the velocity estimates.

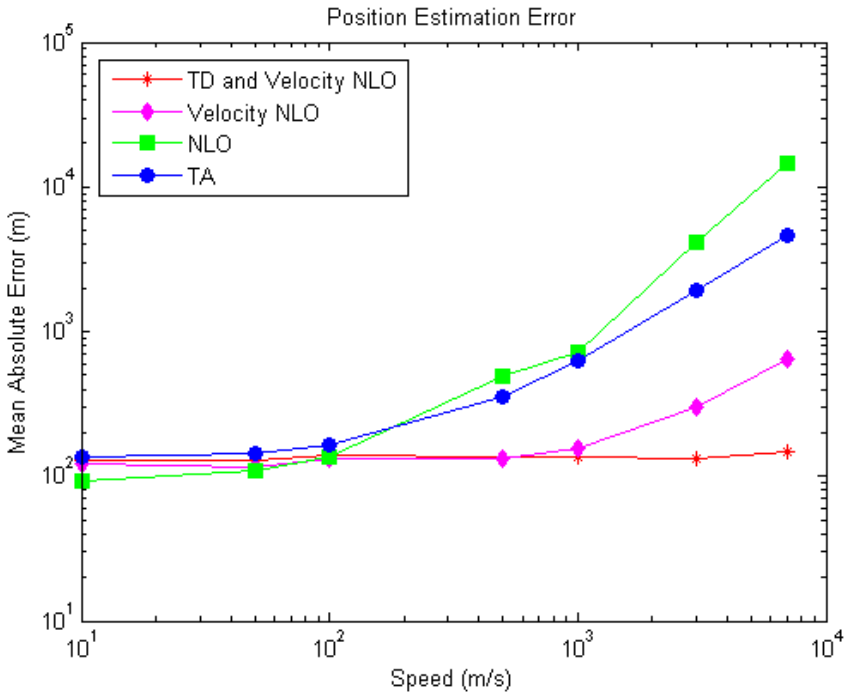

Figure 4.10: The MAE of each geolocation algorithm's position estimates. The time delay and velocity NLO provides constant performance. The other algorithms' accuracies degrade with increased subject velocity.

These effects are also seen in the velocity plot shown in Figure 4.11. As before, the performance of the NLO_{TD} is unaffected by the speed of the subject. All of the other algorithms produce less accurate estimates as the subject's speed increases. The velocity NLO matches the NLO_{TD} MAE much better than it matches position. Now that the geolocation MAE has been addressed for all three NLOs, their NEES is given.

Figure 4.12 shows the NEES of the static and velocity NLO as functions of the subject's velocity. Note that the static NLO and velocity NLOs ideally have NEES values

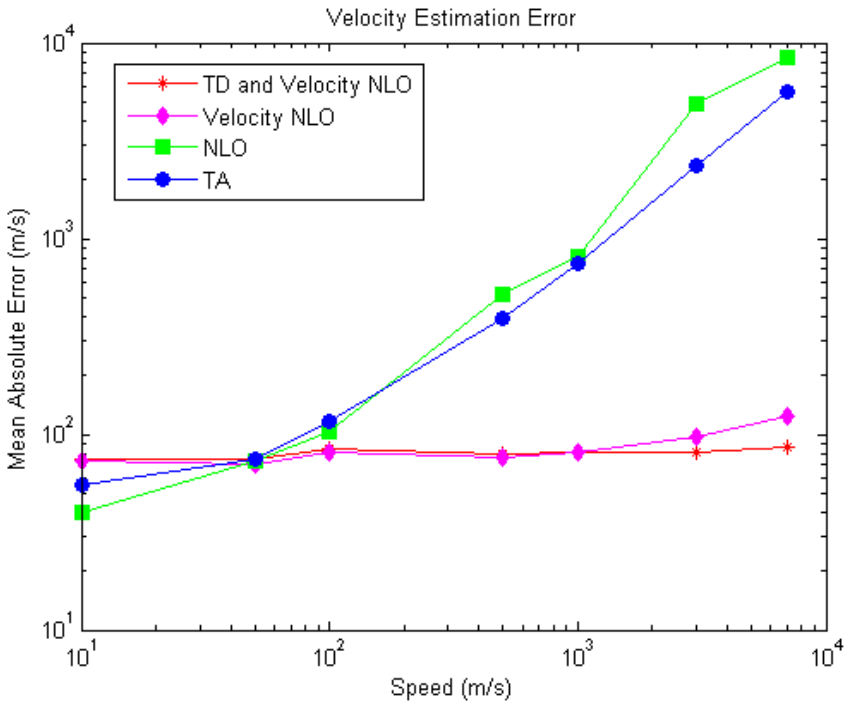

Figure 4.11: The MAE of each geolocation algorithm's velocity estimates. The NLOs that include velocity and/or the time delay provide a nearly constant MAE. The other algorithms' MAEs degrade with increased subject speed.

of 3 and 6, respectively. The static NLO's NEES begins to grow for subjects with speeds of 10 m/s. The velocity NLO provides accurate confidence over a larger range of subject velocities than the static NLO; however, its NEES also grows with the subject's velocity. This indicates that the confidence estimates given by the static and velocity NLOs are overconfident. As expected, the NLO_{TD} was estimated to have a NEES of 6 for all subject velocities.

Because the Velocity and Time Delay NLO includes time delay and the subject's velocity in its model, its estimates should be unbiased. Furthermore, the other NLOs do not include the time delay in their models, so their geolocation estimates may be biased. Therefore, a MCS was performed where LOS measurements of the subject were simulated

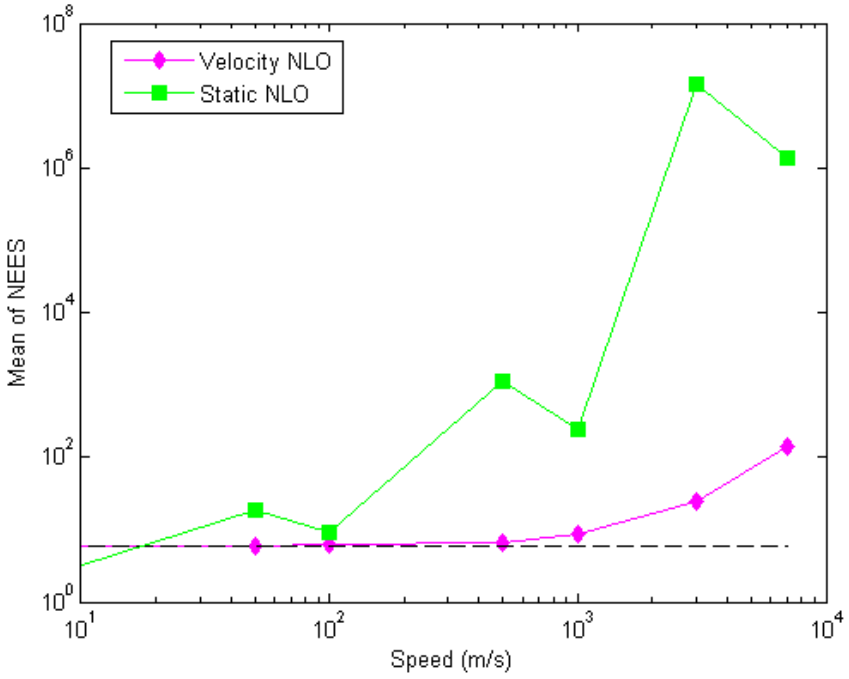

Figure 4.12: The NEES of the static and velocity NLOs for a linear subject with time delay included.

about the true LOS. The NLOs were used with this data to geolocate the subject. The result of this MCS is shown in Figure 4.13.

This section addressed the performance of the NLO_{TD} and the other geolocation algorithms on a subject that moves with constant velocity. This subject was designed to match the assumptions inherent to the NLO_{TD}. The next section describes the performance of the geolocation algorithms on a most challenging subject.

4.4 Relative NLO Performance

In Section 3.1.2, an SV is described as the most challenging subject for geolocation. This subject is anticipated to maximally induces the errors in geolocation due to the subject's path and time delay. The previous three sections describe the performance of the NLOs on simplistic subjects designed to verify the anticipated performance of the NLOs.

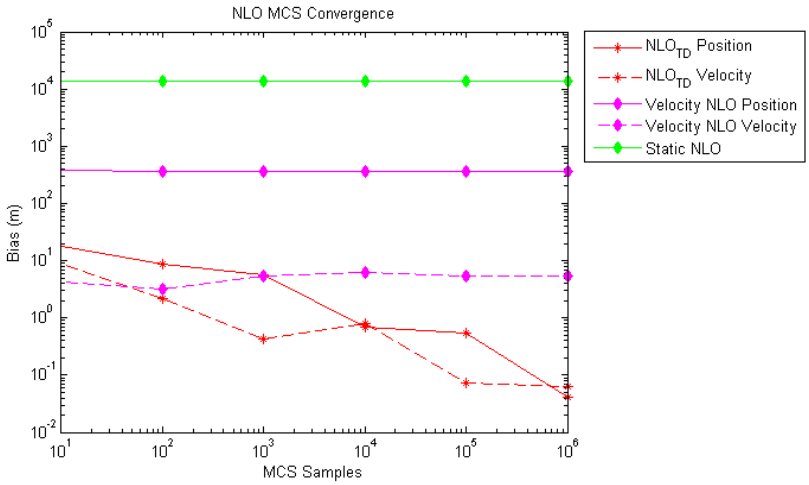

Figure 4.13: The velocity and time delay NLO is unbiased. The other geolocation algorithms produced biased estimates.

This section provides results that will be used to compare the NLOs and TA. A typical simulated SV scenario is shown in Figure 4.14.

Because this section gives results for the most challenging subject, these results give the worst geolocation performance that are are expected for any subject. As discussed in Section 3.2, the LOS measurements were arbitrarily given a standard deviation of error of 5 micro radians. The performance is dependent on the accuracy of the sensors. For example, the accuracy and precision of the geolocation algorithms will decrease if the standard deviation of error in the measurements increases.

To provide the statistics shown in this section, an SV's flightpath is simulated as described in Section 3.1.2. Results are shown for 3, 4, 5, 6, 7, and 8 sensors. A minimum of seven measurements were used from at least 3 independent sensors within a 20 seconds of the estimation time. Twenty different SV scenarios were simulated for each sensor count. Each scenario typically provides more than 1,500 sensors measurements. The subject is estimated at each measurement time. Thus, these statistics are based on approximately 30,000 geolocation estimates.

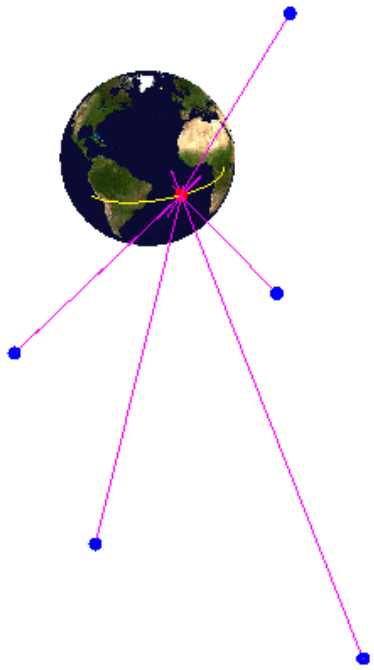

Figure 4.14: A typical SV simulation. The blue points represent satellites. Their measured LOS are shown as the magenta lines. The yellow line is the subject's path, and the point in red is the subject's current position.

Figure 4.15 shows the MAE of the geolocation algorithms as a function of the number of sensors. The staticNLO seems to diverge regardless of the number of sensors present. The TA's estimates have a smaller MAE than the static NLOs, and its MAE decreases with an increased number of sensors. The velocity NLOs estimates are nearly ten times as accurate as the TAs. Its MAE also slowly increases with more sensors. The most advanced NLO, the time delay and velocity NLO produces the most accurate estimates. The MAE of its estimates also increase when using more sensors.

The same effects are seen in the velocity estimates provided by the NLOs. The velocity estimates are shown in Figure 4.16. As before, the static NLO appears to diverge except when there are more than six sensors present. The TA's performance is constant with an increased number of sensors. The velocity NLO and time delay and velocity NLO provide

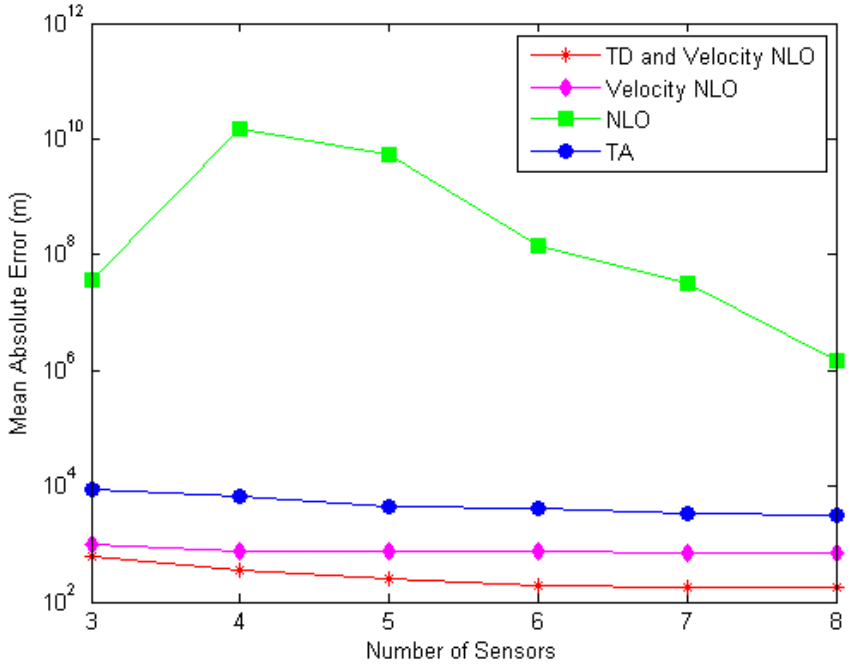

Figure 4.15: The MAE of each geolocation algorithms' position estimates.

similar performance. However, the time delay and velocity NLO provides the best estimate of the subject's velocity for each number of sensors.

Figure 4.17 shows the mean of the NEES for each geolocation algorithm. The static NLO's NEES is much greater than its ideal value of three. This indicates that its confidence estimates are overconfident. The velocity NLO's confidence estimates are also overconfident irrespective of the number of sensors present. The time delay NLO's confidence estimates are also overconfident. However, its error surface estimates become much more accurate as the number of sensors increases. With eight sensors, the time delay and velocity NLO has an NEES of 6.1. Ideally it should have a value of 6, so it is slightly overconfident.

These results were all produced using an arbitrarily chosen sensor confidence. Each sensor's confidence is described by the standard deviation of error in its distributions for

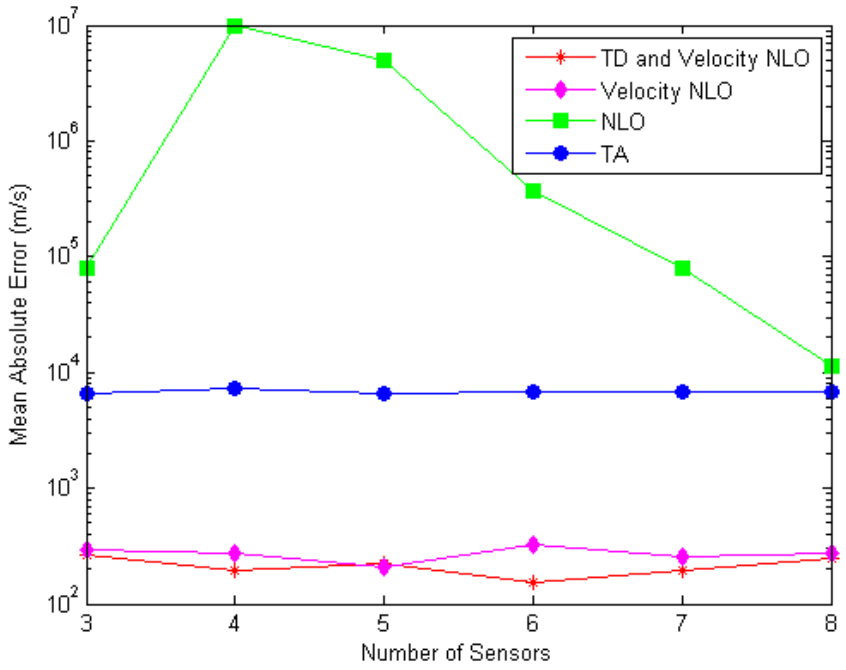

Figure 4.16: The MAE of each geolocation algorithms' velocity estimates.

θ and ϕ. The previously results used a value of 5μR. Figure 4.18 gives the performance of the time delay and velocity NLO as a function of the standard deviation of error in the sensors. Eight sensors were used with the same window settings as the previous results. This data describes how the results scale with the sensor confidence. These results are only shown for the time delay and velocity NLO since it provides the most accurate geolocation estimates for the SV.

The previous discussion concludes the results for this research. This chapter first verified that each NLO matched its theoretical performance in simulations. It also gave information on how velocity and time delay increase the error in geolocation when these are not included in the NLO model. Lastly, results were given detailing the performance of each geolocation algorithm on the most challenging subject. The final chapter will provide conclusions reached from this data, and offer future research recommendations.

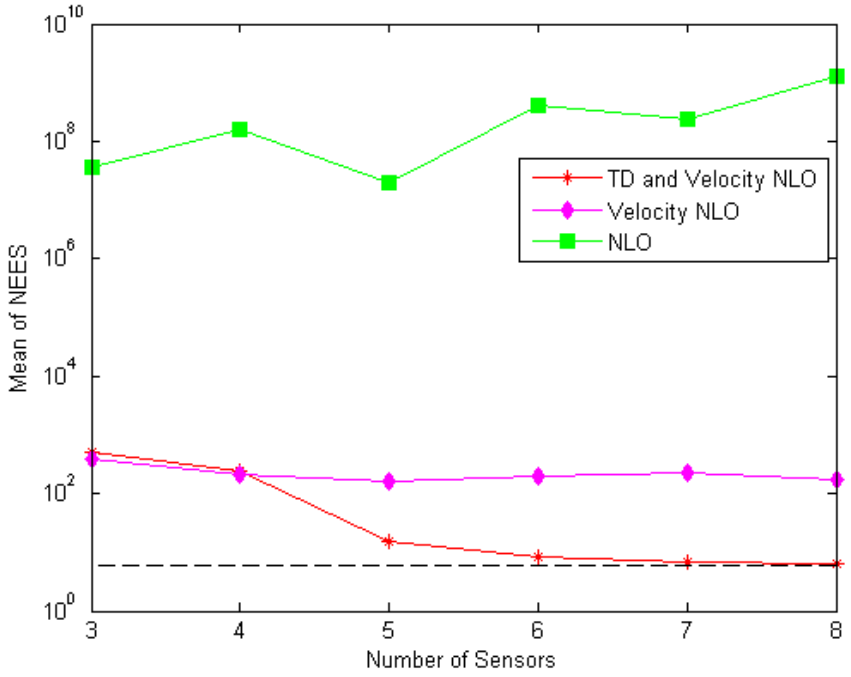

Figure 4.17: The mean of the NEES for each geolocation algorithm's estimates.

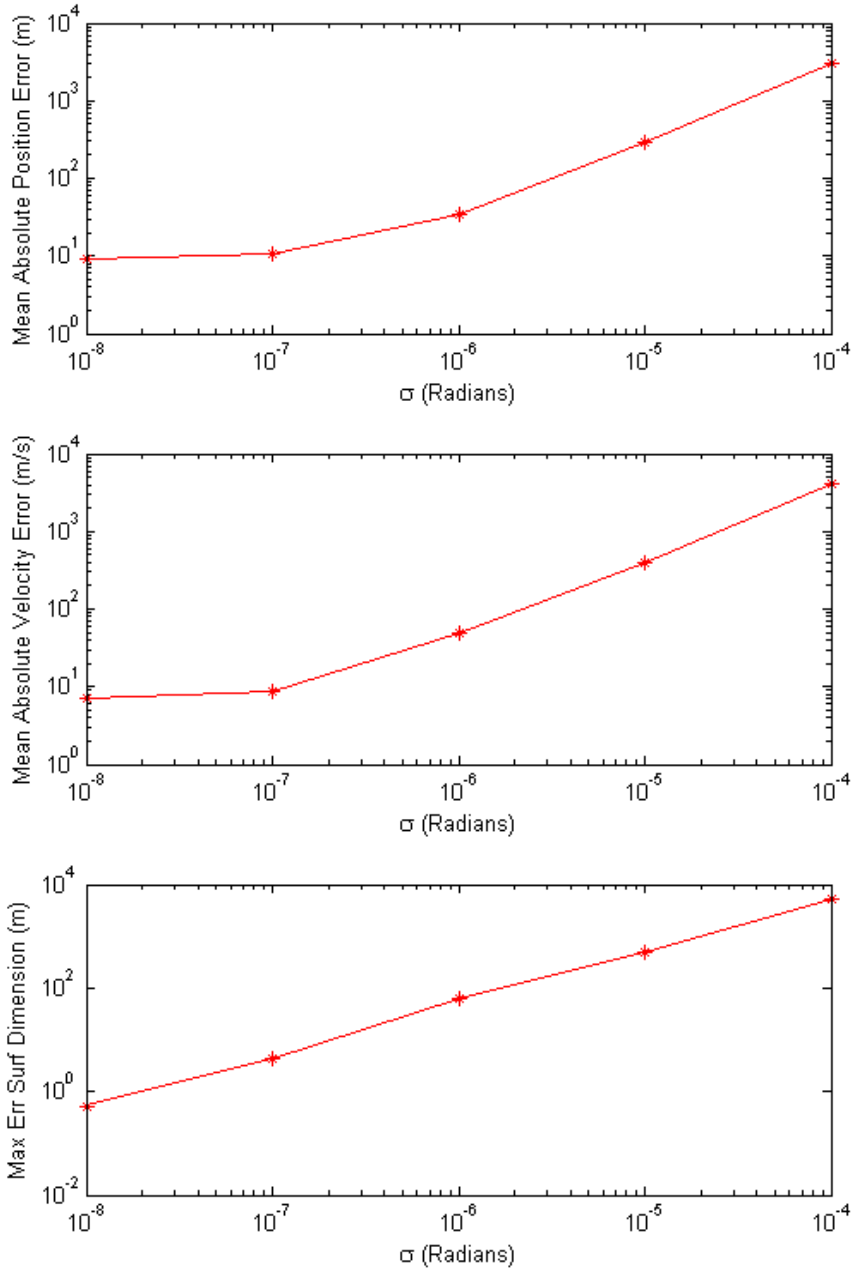

Figure 4.18: The time delay and velocity NLO's performance as a function of sensor confidence.

V. Conclusions & Discussion

In this final chapter, conclusions derived from the results given in the last chapter are discussed. The first section of this chapter describes conclusions regarding theoretical claims about the geolocation algorithms. The next section describes conclusions regarding the magnitude of the errors resulting from geometry, velocity, and time delay. It also discusses the performances of the algorithms when applied to the most challenging subject. Lastly, recommendations and discussion are given for future work.

5.1 Theoretical Verification

In Chapter 2 several claims were made regarding the geolocation algorithms. These theoretical claims were tested via simulations. The motivation for investigating Non-Linear Otpimization (NLO) was to resolve anticipated shortcomings in the Triangulation Algorithm (TA). The TA does not account for system geometry, subject motion, or the time delay.

In Section 4.1, a scenario was simulated with a stationary subject. The TA and static NLO (which includes system geometry in its model) were used to estimate the subject's location. Figure 4.4 gave the accuracy of the geolocation and confidence estimates for the TA and static NLO. It was shown in this section, that the NLO's geolocation estimates have half the MAE provided by the TA's estimates. Furthermore, the NLO provided more confidence in its estimates (its error surfaces were smaller). Both algorithms' metrics improved with more sensors. These results demonstrate that the NLO outperforms the TA by accounting for system geometry.

Furthermore, the static NLO should be an unbiased estimator for a static subject. Thus, the static NLO's average estimate is the true subject location. This claim was tested in

Figure 4.5. The figure demonstrated that the static NLO is an unbiased estimator in the static case.

The static NLO also accurately calculates its error surface (the confidence in its estimates) as illustrated in Figure 4.2. This was also demonstrated by the mean of its NEES. This figure demonstrated that the static NLO's calculated error surface matches the error surface produced by MCS. However, these results should occur only when the static NLO's assumptions are exactly true. The static NLO should not produce ideal results when other factors are included such as the subject's motion and time delay.

The velocity NLO was tested in Section 4.2. In this section, a subject was simulated with constant velocity. The static NLO and TA assume that the subject is stationary while the LOS measurements needed to estimate it are produced. As this approximation becomes less accurate, the static NLO and TAs geolocation estimates have an increased MAE.

This hypothesis was tested in Figures 4.6, 4.7, and 4.8 by implementing the algorithms on subjects moving at different speeds. The results showed that the static NLO out-performed the TA and velocity NLO for subjects with speeds less than 100 m/s. The static NLO out-performs the velocity NLO for slow subjects because the velocity NLO matches its velocity estimate to the random noise.

As the subject's velocity increases, the static NLO's estimates have the largest MAE. This degradation occurs because the static NLO's assumptions become less accurate as the subject's speed increases. It appears that the static NLO may have failed to converge for subjects with speeds greater than 1,000 m/s. As anticipated, the TA's MAE also decreased with speed, although at a slower rate than the static NLO.

The velocity NLO accounts for velocity, so the subject's velocity is innately estimated. For subject's with speeds greater than 100 m/s, the velocity NLO provides the smallest MAE in its geolocation estimates. Its MAE is constant for all subject velocities. The velocity NLO should also be an unbiased estimator. Figure 4.9 demonstrated that the

velocity NLO is an unbiased estimator. Furthermore, the figure showed that the static NLO produces biased geolocation estimates for a moving subject.

The velocity NLO should also accurately estimates the confidence in its estimates. The mean of its NEES was calculated to be six. Because the velocity NLO estimates the position and velocity of the subject in the x, y, and z directions, it estimates six parameters of the subject's state. Therefore, the velocity NLO accurately calculated the error surface (the confidence in its geolocation estimates) in this scenario. However, Figure 4.8 demonstrates that the static NLO over estimates its geolocation confidence, so its error surfaces are not accurate. The static NLOs geolocation estimates are overconfident because the static NLO does not include velocity in its model. Similar results were shown for the time delay and velocity NLO.

Section 4.3 addressed the performance of the time delay and velocity NLO. A subject was simulated with constant velocity and time delay. The error resulting from the time delay increases as the subject's velocity increases; therefore, a faster subject should accentuate this error. The accuracy results for each algorithm was given as a function of the subject's velocity to illustrate this effect. The TA, static NLO, and velocity NLO do not take the time delay into account and so their estimates should become less accurate as the subject's velocity increases.

This effect was observed in Figures 4.10 4.11, and 4.12. As anticipated, the time delay and velocity NLO provides a smaller MAE and improved confidence estimates as the subject's velocity increases compared to the other algorithms. The static NLO performs the best for subject's with speeds less than 100 m/s. However, the static NLO's confidence estimates are overconfident. The velocity NLO also overestimates its confidence by a lesser amount. For subjects with speeds less than 100 m/s, the velocity NLO accurately estimates its error surfaces since its NEES is approximately six over this region. The time delay and

velocity NLO accurately estimates its geolocation confidence irrespective of the subject's speed. The accuracy of its position and velocity estimates is also constant.

Furthermore, the time delay and velocity NLO should be an unbiased estimator in this case. This was addressed by Figure 4.12. The figure demonstrates the time delay and velocity NLO is an unbiased estimator. The other NLOs were biased because they do not include time delay in their models.

The aforementioned results demonstrated the error resulting from the system geometry, subject motion, and time delay. For example, the faster the subject, the more error is produced in geolocation estimates. This may give validity to the assumption that an SV is the most challenging subject for geolocation. An SV is the fastest subject that might be geolocated and its motion is complex (meaning it accelerates and has other higher order moments). Therefore, it could be reasonably expected to produce the most error in geolocation estimates. The application of these geolocation algorithms to such a subject is discussed in the next section.

5.2 Algorithm Comparison

The ultimate test of the geolocation algorithms considered here was to apply them against the most challenging subject: a Space Vehicle (SV). Section 4.4 describes the results of applying these algorithms to geolocate simulated SVs. Figures 4.15 and 4.16 demonstrated that the static NLO does not converge to the estimate nearest to subject's position for SVs. This divergence indicates that its assumptions are too inaccurate to geolocate such a subject. The TA's estimates have a larger MAE than the velocity and time delay and velocity NLOs. The TA's MAE slightly improves with an increased number of sensors.

The MAE of the velocity NLO's estimates is nearly one hundred times smaller than the MAE for the TA's estimates. The time delay and velocity NLO provides the smallest MAE in position and velocity. These results were produced for a sensor confidence of

5μR. Figure 4.18 provides results for the time delay and velocity NLO with other sensor confidences. As anticipated, more accurate sensors result in smaller MAE in geolocation estimates and improved confidence. Thus changing sensor confidence results in a vertical shift in the geolocation accuracy plots given in this research.

Perhaps most important, the time delay and velocity NLO's NEES nearly converges to its ideal value for a sufficient number of sensors. Figure 4.17 demonstrates that none of the other geolocation algorithms converge to their ideal NEES value. Therefore, the time delay and velocity NLO provides the most accurate geolocation confidence estimates for the most challenging subject.

While the static NLO could not provide useful estimates for the most challenging subject, its estimates have the smallest MAE for the position and velocity of slow subjects. Therefore, if the type of subject being observed is known, then it may be better to use the static NLO. For example, if the user knows they are observing a static subject, then the static NLO may provide better results.

The MAE of the velocity NLO's estimates is larger than the static NLO's MAE for low-speed subjects and larger than the time delay and velocity NLO's MAE for high-speed subjects. Therefore, the velocity NLO does not appear to be useful. For unknown subjects, it may be best to use the time delay and velocity NLO because it is the most robust algorithm provided here and gives the most accurate confidence estimates of the geolocation algorithms.

The previous discussion answers the first part of the research question: which NLO performs the best? The static NLO provides the best geolocation algorithm for subjects with speed less than 10 m/s. The time delay and velocity NLO provides the best geolocation performance for general subjects. The second part of the research question is how confidence in estimates might be visualized. A visualization of geolocation confidence was provided by the error surfaces. This research also demonstrated that the error surfaces

can accurately depict the confidence in geolocation estimates. However, none of the algorithms could accurately estimate the confidence in their geolocation estimates for the most challenging subject. The research question has been addressed; however, there are still remaining problems that may be addressed in follow-on research.

5.3 Future Work

As described in Chapter 1, there were many assumptions and limitations for this research. Perhaps the most significant of the assumptions is that the measurements are unbiased. If LOS measurements are biased, then none of the results given here will hold. In this case, the NLOs will all over-estimate their confidences. Depending on the magnitude of the bias, the bias could be a larger source of error than the random jitter in the measurements.

Furthermore, the sensors in this research measured the LOS with a pseudo-random periodicity. Measurement times may not be consistent in reality. The measurement times were also assumed to be perfectly known. In reality, there may be uncertainty in the measurement times. It may be valuable to analyze the effects of adding more realism to the sensor measurement times on geolocation. Relativistic effects could also be considered.

The effects of different sensors on geolocation have not been addressed. For example, it may be valuable to consider the accuracy and confidence improvement resulting from adding one sensor close to the subject. The accuracy and confidence in geolocation estimates could be calculated as a function of sensor distance to the subject.

Lastly, the effects of filtering the LOS data could be useful. While the noise in the LOS measurements is independent, the measurements are still taken of the same subject. Therefore, one LOS measurement provides information on the next measurement. A Kalman filter could be used to improve the LOS measurements. Something akin to this began to be examined; however, it was mistakenly performed. A discussion of this investigation is given in Appendix C.

These are just some of the ideas that arose over the course of this research. The conclusions and discussion provided here give closure to the research question, and have resulted in potential queries for further research.

Appendix A: Code

A.1 SV_PathGen.m

```
function [ sv ] = sv_PathGen( sv )
%% Authorship
%    Stephen Hartzell, AFIT - GRA
%    stephen.hartzell.ctr@afit.edu, hartzell.stephen@gmail.com
%    Created: 06-01-2011, Modified: 10-27-2012
%
% sv_PathGen Generate or update an space vehicle (SV) flightpath
%    [ sv ] = sv_PathGen( sv ) performs one of two operations
%    depending on the format of the input sv. If input sv is not a
%    structure, then realistic parameters that completely define ...
    an sv's
%    path are generated. If input sv is a structure with the required
%    elements, then the subject's positions and velocities at the ...
    times
%    defined by sv.t are generated for an sv's flight path in a
%    2D-plane. The resulting sv is then rotated according to a ZXY
%    rotation and the new 3D points are stored in sv.r structure. The
%    full sv structure will have up to fifteen <1x1> structure ...
    fields:
%    a, Alpha, b, Beta, C_e, Omega, Rot_X, Rot_Y, Rot_Z, t, N, r, v,
%    t_reentry_max, and t_max
%
%    N is a vector pointing from center of the Earth towards a ...
    plane which it
%    is normal to. The sv's flight path will be generated such ...
    that the
%    plane containing the flight path is orthogonal to the plane ...
    which N is
%    orthogonal to. N is useful if the sv flight path must be ...
    located such that
%    it is observable by a system of satellites.
%
%    NOTE 1: The only known problem with this generator is that ...
    for large
%    enough ArcAngles, the generated path is unrealistic and ...
    results in a
%    much longer flight duration than is realistic.
%
%    NOTE 2: Within the function, v_f provides an approximation ...
    of the
%    maximum velocity which the subject will achieve at any time ...
    along its
%    path. Maximum velocities will vary, but will always stay within
%    reasonable bounds (in all test cases: below 10 Km/s). A very ...
    unique
```

```
%    combination of inputs would be required to produce a ...
    velocity of this
%    magnitude or greater, and it may not be possible.
%

%% Define constants

% Check to see if a time series is included. If it isn't, create ...
    a new
% scenario. If it is, then generate positions based on times
if isstruct(sv)
    % sv is already populated
        [ sv ] = generate_data( sv );
else
    % sv needs to be populated
        [ sv ] = generate_sv();
end
end

function sv = generate_sv()
% Generate the data needed to get sv positions at any time

% Unit vector normal to a plane which is tangent to Earth
N = randn(3,1);
N = N./norm(N);

% Radius of the Earth
r = 6378100;

% The number below is the minimum arc angle for which an sv would ...
    be used
% to hit a subject. The full definition of ArcAngle is given as a ...
    comment
% next to the used definition.
ArcAngle = .862325770997633+2.279266882592160*rand; %ArcAngle = ...
    5500000/r+(pi-5500000/r)*rand

Apogee = 900000+rand*600000;

%% Define Ellipse

% Y coordiante of the center of the ellipse
C_e = r-2800000;

% The constant which scales the y-components of the ellipse
a = Apogee + r - C_e;

% This determines the starting point of the subject's path
Beta = asin((C_e-r*cos(ArcAngle/2))/a);

% The constant which scales the x-components of the ellipse
b = r*sin(ArcAngle/2)/cos(Beta);
```

```
81
82 %% Reentry Phase
83
84 % The final altitude of the sv during the reentry phase
85 a_f = 150000+300000*rand;
86
87 % The velocity at the end of the reentry phase
88 v_f = 5000+2000*rand;
89
90 Alpha_1 = ...
       asin((-a*C_e+sqrt((a*C_e)^2-(a^2-b^2)*(C_e^2+b^2-(a_f+r)^2))) ...
91      / (a^2-b^2));
92 Alpha_2 = ...
       asin((-a*C_e-sqrt((a*C_e)^2-(a^2-b^2)*(C_e^2+b^2-(a_f+r)^2))) ...
93      / (a^2-b^2));
94
95 % Ensure that the correct value of alpha is chosen given the two ...
       possible
96 % values.
97 if (isreal(Alpha_1) == 0) && (isreal(Alpha_2) == 0)
98     error('Illegitimate sv Generation')
99 elseif isreal(Alpha_1) && isreal(Alpha_2)
100     if alpha_1 < alpha_2
101         Alpha = Alpha_1;
102     else
103         Alpha = Alpha_2;
104     end
105 elseif isreal(Alpha_1) == 0
106     Alpha = Alpha_2;
107 else
108     Alpha = Alpha_1;
109 end
110
111 omega = v_f / (sqrt((b*sin(Alpha))^2+(a*cos(Alpha))^2) * 2);
112
113 % The length of time it takes to complete the reentry phase
114 t_reentry_max =  (Alpha+Beta)/omega;
115
116 % Time samples across the orbital-phase
117 t_max = (pi+2*Beta)/(2*omega)+omega*t_reentry_max;
118
119 sv.t_max = t_max;
120 sv.t_reentry_max = t_reentry_max ;
121 sv.a = a;
122 sv.b = b;
123 sv.Alpha = Alpha;
124 sv.Beta = Beta;
125 sv.C_e = C_e;
126 sv.omega =omega;
127 sv.Rot_X =  atan2(N(3),norm(N(1:2)));
128 sv.Rot_Y = rand*pi;
129 sv.Rot_Z = -atan2(N(1),N(2));
```

```
130 sv.N = N;
131
132 end
133
134 function    [ sv] = generate_data(sv)
135 % Use the values in sv to generate positions & velocities at ...
      times given by sv.t
136
137 reentry_length = sum(sv.t < sv.t_reentry_max);
138
139 t = sv.t(1:reentry_length);
140 sv_r = -sv.r.*cos((t./sv.t_reentry_max).*sv.omega.*t-sv.Beta);
141 sv_rv = 2 * sv.r * sv.omega .* t .* sin((t./sv.t_reentry_max).*...
142     sv.omega.*t-sv.Beta) ./sv.t_reentry_max;
143 sv_r(2,:) = sv.a.*sin((t./sv.t_reentry_max).*...
144     sv.omega.*t-sv.Beta)+sv.C_e;
145 sv_rv(2,:) = 2 * sv.a * sv.omega .* t .*...
146     cos((t./sv.t_reentry_max).*sv.omega.*t-sv.Beta) ...
          ./sv.t_reentry_max;
147 %% Orbital Phase
148
149 t = sv.t(reentry_length + 1:end);
150 sv_o = -sv.b.*cos(2.*sv.omega.*t-sv.Beta-sv.omega*...
151     sv.t_reentry_max);
152 sv_ov = 2*sv.b*sv.omega.*sin(2.*sv.omega.*t-sv.Beta-...
153     sv.omega*sv.t_reentry_max);
154 sv_o(2,:) = sv.a.*sin(2.*sv.omega.*t-sv.Beta-sv.omega*...
155     sv.t_reentry_max)+sv.C_e;
156 sv_ov(2,:) = 2*sv.a*sv.omega.*cos(2.*sv.omega.*t-sv.Beta...
157     -sv.omega*sv.t_reentry_max);
158
159 %% Add Phases
160
161 svp = sv_r(1,:);
162 svp(2,1:size(sv_r,2)) = sv_r(2,:);
163 svp(1,(size(sv_r,2)+1):(size(sv_r,2)+size(sv_o,2)))...
164     = sv_o(1,:);
165 svp(2,(size(sv_r,2)+1):(size(sv_r,2)+size(sv_o,2)))...
166     = sv_o(2,:);
167 svp(3,:) = zeros(1,size(svp,2));
168
169 svv = sv_rv(1,:);
170 svv(2,1:size(sv_rv,2)) = sv_rv(2,:);
171 svv(1,(size(sv_rv,2)+1):(size(sv_rv,2)+size(sv_ov,2)))...
172     = sv_ov(1,:);
173 svv(2,(size(sv_rv,2)+1):(size(sv_rv,2)+size(sv_ov,2)))...
174     = sv_ov(2,:);
175 svv(3,:) = zeros(1,size(svv,2));
176
177 sv.r = ZXY_Rotation(svp,sv.Rot_Z,sv.Rot_X,sv.Rot_Y);
178 sv.v = ZXY_Rotation(svv,sv.Rot_Z,sv.Rot_X,sv.Rot_Y);
179 end
```

A.2 OrbGen.m

```
function [ Sat ] = OrbGen( N, t )
%% Authorship
%    Stephen Hartzell, AFIT - GRA
%    stephen.hartzell.ctr@afit.edu, hartzell.stephen@gmail.com
%    Created: 02-04-2011, Modified: 06-06-2012
%
% OrbGen Generate satellite orbital paths.
%    [ SAT ] = OrbGen( N, T ) generates a psuedo random realistic ...
    satellite
%    orbit on the opposite side of an Earth-tangental plane ...
    defined by N.
%    N is a unit vector which points from the center of mass of ...
    the Earth
%    towards a plane which is tangential to the Earth. T is an array
%    containing times at which satellites positions are to be ...
    generated.
%    Positions saved to SAT correspond to the times in T are ...
    stored as
%    column vectors in a matrix with the format [Sx; Sy; Sz].
%
%    Note: T is restricted less it become impossible to generate ...
    an orbital
%    section only on the side of the plane opposite the Earth. ...
    For this
%    reason, LEO satellites are not generated, since their ...
    orbital periods
%    are comparable to the flight length of the space vehicle.

%% Generate Coherent Sections of Orbits Defined By Angular Region

% The check is here to guard against the rare (and perhaps
% impossible) situation of orbit parameters along with N causing
% the function to be unable to find a realistic satellite section
chk = 0;
while chk == 0
    switch randi([2,4])
        case 1
            % LEO
            LEO.name = 'LEO';
            LEO.eccentricity = .1*rand;
            LEO.SemiMajAx = 7378000; % m
            LEO.SemiMinAx = LEO.SemiMajAx * ...
                sqrt(1-LEO.eccentricity^2); % m
            LEO.inc = rand * 90 * pi/180;
            LEO.LoAN = rand * 360 * pi/180;
            LEO.AoP = rand * 90 * pi/180;
            LEO.Vt = 7800; % m/s
            LEO.Circumference = pi*(LEO.SemiMajAx+LEO.SemiMinAx)...
                *(1 + ((3*((LEO.SemiMajAx-LEO.SemiMinAx)/...
```

```
                (LEO.SemiMajAx+LEO.SemiMinAx))^2)/(10 + sqrt(4-3*...
                ((LEO.SemiMajAx-LEO.SemiMinAx)/...
                (LEO.SemiMajAx+LEO.SemiMinAx))^2))));
            LEO.TrueAn = rand * 360 * pi/180;
            LEO.Period = LEO.Circumference / LEO.Vt; % s
            I = LEO;
        case 2
            % MEO
            MEO.name = 'MEO';
            MEO.eccentricity = .05*rand;
            MEO.SemiMajAx = 20200000; % m
            MEO.SemiMinAx = MEO.SemiMajAx * ...
                sqrt(1-MEO.eccentricity^2); % m
            MEO.inc = rand * 90 * pi/180;
            MEO.LoAN = rand * 360 * pi/180;
            MEO.AoP = rand * 90 * pi/180;
            MEO.Vt = 4000; % m/s
            MEO.Circumference = ...
                pi*(MEO.SemiMajAx+MEO.SemiMinAx)*(1 +...
                ((3*((MEO.SemiMajAx-MEO.SemiMinAx)/(MEO.SemiMajAx+...
                MEO.SemiMinAx))^2)/(10 + sqrt(4-3*((MEO.SemiMajAx-...
                MEO.SemiMinAx)/(MEO.SemiMajAx+MEO.SemiMinAx))^2))));
            MEO.TrueAn = rand * 360 * pi/180;
            MEO.Period = MEO.Circumference / MEO.Vt; % s
            I = MEO;
        case 3
            % GEO
            GEO.name = 'GEO';
            GEO.eccentricity = .1*rand;
            GEO.SemiMajAx = 42164000; % m
            GEO.SemiMinAx = GEO.SemiMajAx * ...
                sqrt(1-GEO.eccentricity^2); % m
            GEO.inc = rand * 90 * pi/180;
            GEO.LoAN = rand * 360 * pi/180;
            GEO.AoP = rand * 90 * pi/180;
            GEO.Vt = 3075; % m/s
            GEO.Circumference = ...
                pi*(GEO.SemiMajAx+GEO.SemiMinAx)*(1 +...
                ((3*((GEO.SemiMajAx-GEO.SemiMinAx)/(GEO.SemiMajAx+...
                GEO.SemiMinAx))^2)/(10 + sqrt(4-3*((GEO.SemiMajAx-...
                GEO.SemiMinAx)/(GEO.SemiMajAx+GEO.SemiMinAx))^2))));
            GEO.TrueAn = rand * 360 * pi/180;
            GEO.Period = GEO.Circumference / GEO.Vt; % s
            I = GEO;
        case 4
            % HEO
            HEO.name = 'HEO';
            HEO.eccentricity = .2*rand;
            HEO.SemiMajAx = 60000000; % m
            HEO.SemiMinAx = HEO.SemiMajAx * ...
                sqrt(1-HEO.eccentricity^2); % m
            HEO.inc = rand * 90 * pi/180;
```

```
88              HEO.LoAN = rand * 360 * pi/180;
89              HEO.AoP = rand * 90 * pi/180;
90              HEO.Vt = 2600; % m/s
91              HEO.Circumference = ...
                    pi*(HEO.SemiMajAx+HEO.SemiMinAx)*(1 +...
92                   ((3*((HEO.SemiMajAx-HEO.SemiMinAx)/(HEO.SemiMajAx+...
93                   HEO.SemiMinAx))^2)/(10 + sqrt(4-3*((HEO.SemiMajAx-...
94                   HEO.SemiMinAx)/(HEO.SemiMajAx+HEO.SemiMinAx))^2))));
95              HEO.TrueAn = rand * 360 * pi/180;
96              HEO.Period = HEO.Circumference / HEO.Vt; % s
97              I = HEO;
98      end
99
100     I.ArcAng = (2*pi*max(t))/I.Period;
101
102     % Define angle between V and N, V defines the orientation of the
103     % y-axis of the plane containing the generated ellipse
104     if strcmp(I.name,'LEO')
105         ang = rand*15*pi/180;
106     elseif strcmp(I.name,'MEO')
107         ang = rand*45*pi/180;
108     elseif strcmp(I.name,'GEO')
109         ang = rand*65*pi/180;
110     elseif strcmp(I.name,'HEO')
111         ang = rand*65*pi/180;
112     end
113
114     % Generate random vector U and find the the cross product of ...
          N and
115     % U to create a unit vector which over-writes U which is ...
          orthogonal
116     % to N.
117     U = randn(3,1);
118     U = cross(U,N);
119     U = U ./ norm(U);
120
121     % Create vector V which defines the orientation of the elliptical
122     % orbits
123     U = 6378000.*tan(ang).* U;
124     V = 6378000.*N+U;
125
126     % The angles which define the orientation of the elliptical ...
          orbits
127     I.LoAN = -atan2(V(1),V(2));
128     I.inc = atan2(V(3),norm(V(1:2)));
129     I.AoP = rand*pi;
130
131     %% Find Angular Limits for Arguments of the Ellipse's Defintion
132
133     % Initialize values
134     alpha = 0;
135     d_alpha = .2;
```

```
136     dist = 2;
137
138     % This finds the positive angle, alpha, which if input into the
139     % equation of the ellipse in its own plane corresponds to one of
140     % the two points at which the ellipse intersects the equation of
141     % the plane which is tangent to the Earth (as defined by N). This
142     % loop finds alpha by an iterative approach.
143     while dist > 1e-6
144
145         % Try a new value of alpha
146         alpha = alpha + ∆_alpha;
147
148         % Define position of ellipse in 2D plane at alpha. The
149         % convention of defining the ellipse as shown below:
150         % x-values = -sin and y-values = -cos, was chosen to ensure
151         % that for alpha = 0, P_r is the point on the ellipse
152         % opposite V (see V above)
153         Px = -I.SemiMajAx*sin(alpha);
154         Py = -I.SemiMinAx*cos(alpha);
155
156         % Rotate the previously defined positions along the ellipse
157         % in its own plane to define the points P_r in 3D space
158         % which correspond to the points in the ellipses own plane
159         [ P_r ] = ZXY_Rotation( [Px;Py;0], I.LoAN, I.inc, I.AoP );
160
161         % ang is overwritten and now contains the angle between the
162         % vector pointing from the center of mass of the Earth to P_r
163         % and N
164         ang = acos((P_r./norm(P_r))'*N);
165
166         if ang > pi/2
167             last_alpha = alpha;
168         elseif ang < pi/2
169             ∆_alpha = ∆_alpha/2;
170
171             % Saves the last angle which did not go on the other
172             % side of the plane
173             alpha = last_alpha;
174         end
175         dist = abs(ang-pi/2);
176     end
177
178     % Initialize values
179     beta = 0;
180     ∆_beta = -.2;
181     dist = 2;
182
183     % This finds the negative angle, beta, which if input into the
184     % equation of the ellipse in its own plane corresponds to one
185     % of the two points at which the ellipse intersects the
186     % equation of the plane which is tangent to the Earth (as
187     % defined by N). This loop finds beta by an iterative approach.
```

```
188     while dist > 1e-6
189
190         % Try a new value of beta
191         beta = beta + ▵_beta;
192
193         % Define position of ellipse in 2D plane at alpha. The
194         % convention of defining the ellipse as shown below:
195         % x-values = -sin and y-values = -cos, was chosen to ensure
196         % that for alpha = 0, P_r is the point on the ellipse
197         % opposite V (see V above)
198         Px = -I.SemiMajAx*sin(beta);
199         Py = -I.SemiMinAx*cos(beta);
200
201         % Rotate the previously defined positions along the ellipse
202         % in its own plane to define the points P_r in 3D space
203         % which correspond to the points in the ellipses own plane
204         [ P_r ] = ZXY_Rotation( [Px;Py;0], I.LoAN, I.inc, I.AoP );
205
206         % ang is overwritten and now contains the angle between the
207         % vector pointing from the center of mass of the Earth to P_r
208         % and N
209         ang = acos((P_r./norm(P_r))'*N);
210
211         if ang > pi/2
212             last_beta = beta;
213         elseif ang < pi/2
214             ▵_beta = ▵_beta/2;
215
216             % Saves the last angle which did not go on the other
217             % side of the plane
218             beta = last_beta;
219         end
220         dist = abs(ang-pi/2);
221     end
222
223     % The angular region in which the satellite's orbit can exist
224     max_ang = 2*pi - (alpha-beta);
225
226     if max_ang < I.ArcAng
227         chk = 0;
228     else
229         chk = 1;
230
231         % Define TrueAn such that the orbit section will exist on
232         % the side of the plane pointed to by N and is randomly
233         % positioned on that side
234         I.TrueAn = alpha + rand * (max_ang - I.ArcAng);
235
236         % Angular velocity
237         Aw = 2*pi / I.Period;
238
239         % Define the motion of the satellite in its own plane
```

```
240         Px = -I.SemiMajAx*sin(Aw.*t + I.TrueAn);
241         Py = -I.SemiMinAx*cos(Aw.*t + I.TrueAn);
242         Pz = zeros(1,length(Px(1,:)));
243 
244         % Rotate the previously defined positions along the ellipse
245         % in its own plane to define the points P_r in 3D space
246         % which correspond to the points in the ellipses own plane
247         [ P_r ] = ZXY_Rotation( [Px;Py;Pz], I.LoAN, I.inc, I.AoP );
248 
249         Sat(1,(1:size(P_r,2))) = P_r(1,:); %#ok<*AGROW>
250         Sat(2,(1:size(P_r,2))) = P_r(2,:);
251         Sat(3,(1:size(P_r,2))) = P_r(3,:);
252         Sat(4,1:length(t)) = t;
253     end
254 end
255 end
```

Appendix B: Frames

As discussed in Section 2.4, the NLO may converge to any one of several solutions to the nonlinear system of equations. The NLO relies on the assumption that the measurement space as a function of the subject's state $\mathbf{\Omega}(\mathbf{x})$ is approximately linear near the initial guess at the solution. Early in this research, it was thought that this linear approximation would be more accurate if the angular measurements were within $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$. This reasoning is no longer considered valid. However, for this reason, the NLOs were written to use four different coordinate systems.

It was thought to be desirable to make the $\mathbf{\Omega}(\mathbf{x})$ function approximately linear. From Equation (3.9) it can be seen that the LOS measurements depend on an arctangent function. From Figure B.1 it is evident that the arctangent function is well approximated by a line with a slope of one on the interval $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$. However, the linear approximation may be inaccurate in the wings.

Therefore, a set of four coordinate systems are used to keep the measurement angles within $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$. These coordinate systems are called *frames*.

If the LOS measurements fall outside of the desirable interval, then the measurements are converted to another frame, and the optimization operates within that frame. The frames are defined as shown below.

The frames in which the measurements are placed are determined as follows. The measurements are converted to frame

1. if $-\frac{\pi}{4} \leq \phi \leq \frac{\pi}{4}$ and $-\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}$ or $\frac{3\pi}{4} \leq \theta \leq \frac{5\pi}{4}$
2. if $-\frac{\pi}{4} \leq \phi \leq \frac{\pi}{4}$ and $\frac{\pi}{4} < \theta < \frac{3\pi}{4}$ or $\frac{-3\pi}{4} < \theta < \frac{-\pi}{4}$
3. if $|\phi| > \frac{\pi}{4}$ and $-\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}$ or $\frac{3\pi}{4} \leq \theta \leq \frac{5\pi}{4}$

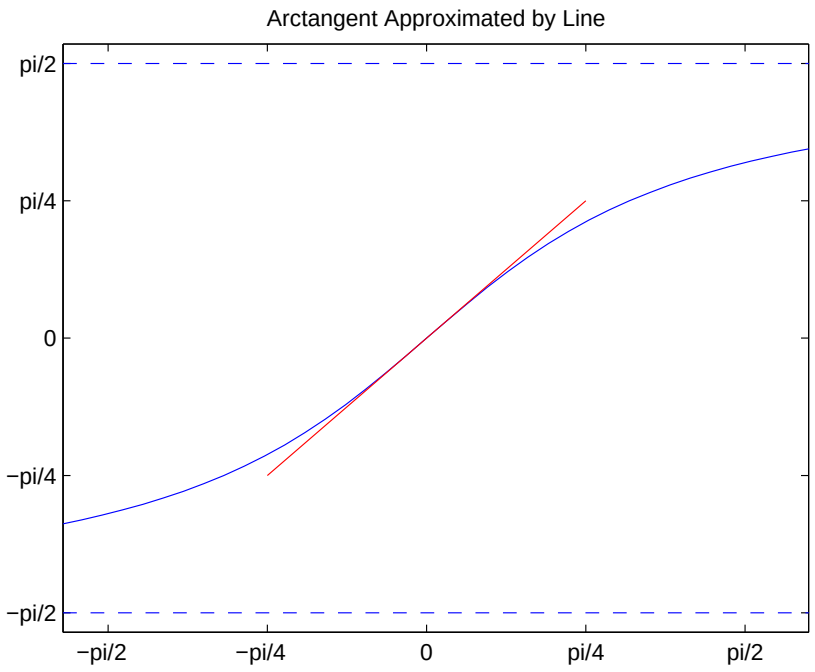

Figure B.1: Linear approximation of arctangent function. The arctangent function is reasonably approximated by a line with a slope of one over the domain $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$.

4. if $|\phi| > \frac{\pi}{4}$ and $\frac{\pi}{4} < \theta < \frac{3\pi}{4}$ or $\frac{-3\pi}{4} < \theta < \frac{-\pi}{4}$

All data are synthesized in frame 3. When using the NLO, each measurement is converted to the appropriate frame.

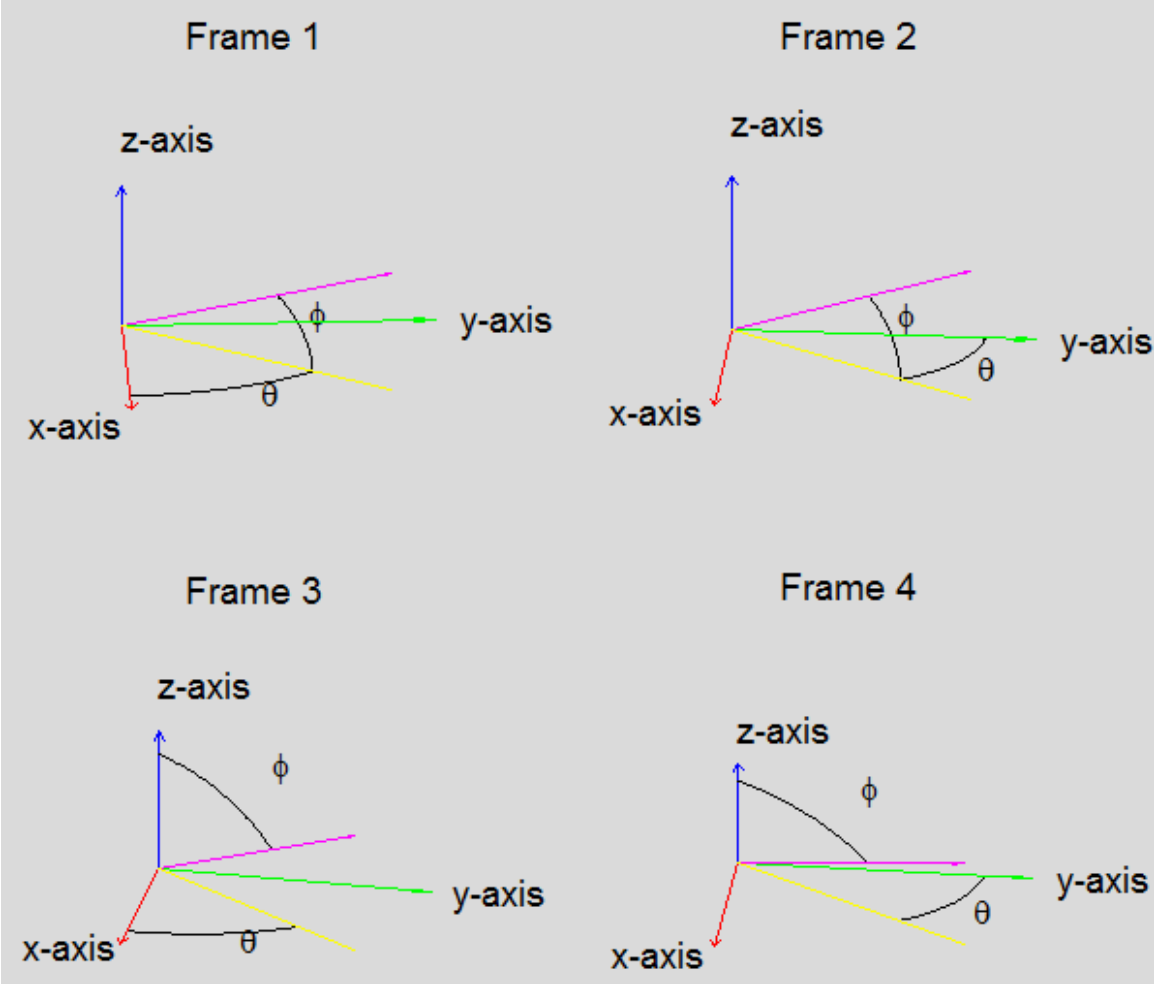

Figure B.2: Frame definitions. The four frames used for measurement data when performing the NLO.

Appendix C: Extended Kalman Filter

This appendix represents additional research that was conducted to improve geolocation estimates. This approach was determined to be inappropriate because it assume that estimates of the subject's state are independent. A LOS measurement may be used for more than one estimation. Therefore, subject estimates are not independent. However, there may still be some utility to this investigation, and so it is included here in the case of future inquiry.

Sensors may take measurements of the subject in question over long periods of time. Therefore, when multiple sensors have taken measurements at nearly the same time, an estimate of the subject's position may be produced. Therefore, the subject's track will be sampled by geolocation estimates. This is shown in Figure C.1. The TA and NLO considers each geolocation estimate independently. Therefore, the subject's track is never taken into consideration by the NLO. This is information could be used to improve the geolocation estimates and confidence estimates.

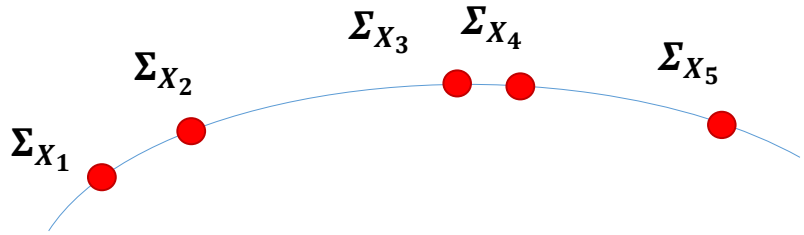

Figure C.1: A track of a geolocation estimates each with their own covariance matrices. There is an apparent track; however, neither the TA nor the NLO natively take this track into account.

If an unknown subject is being observed, it's track is likely unknown; however, over short lengths of time, a subject's track may be accurately described by a second order

model where acceleration is constant. Uncertainty in this model may be described by its own covariance matrix. Therefore, an geolocation estimate may be projected forward or backward by a model to estimate the subject at another time. The covariance matrix for this new projected estimate depends on the confidence in the original estimate and the model. The Kalman Filter uses this scheme to improve geolocation estimates at all points in time [22]. First this section will discuss how to predict and refine and a geolocation estimate using a previous estimate. The discussion will then be expanded to also retrodict from the subsequent estimate. This development is similar to that given in [22].

C.1 The Linear Estimation Problem

The linear estimation problem involves attempting to produce the ideal estimate of a subject's state given an imperfect model of the subject's motion and noisy measurements of this subject's state. The subject's motion is described as

$$\mathbf{x}_{k+1} = \vec{\mathbf{\Phi}}_k \mathbf{x}_k + \mathbf{G}_k \mathbf{U}_k \tag{C.1}$$

In this equation, $\vec{\mathbf{\Phi}}_k$ is the called the transition matrix. The right arrow represents that it transforms the subject's state forward to $k + 1$. It describes the subject's motion if the model were perfect. $\mathbf{G}_k$ describes the effect of noise in the model on the subject's next state. For our purposes, $\mathbf{G}_k$ is the identity matrix. $\mathbf{U}_k$ is a vector of zero-mean noise with covariance matrix $\mathbf{\Sigma_\Phi}$. k is an indexing term.

The measurements $\mathbf{\Omega}_k$ associated with the subject at $\mathbf{x}_k$ are described as

$$\mathbf{\Omega} = \mathbf{M}_k \mathbf{x}_k + \mathbf{N}_k \tag{C.2}$$

In this equation, $\mathbf{M}_k$ is a matrix that transforms the subject's state into measurements. $\mathbf{N}_k$ is a noise vector with covariance matrix $\mathbf{\Sigma_\Omega}$. This model will be used to predict the state and variance at $k + 1$.

C.2 Prediction

The best estimate of the true state at k is $\hat{\mathbf{x}}_k = \mathbf{x}_k + \epsilon_k$ where ϵ_k is the error in the estimate. The error is mean-zero and has a covariance matrix of $\mathbf{\Sigma}_{\mathbf{x}_k}$. Since $\mathbf{U}_k$ is mean-zero, the best prediction $\overrightarrow{\mathbf{x}}_{k+1}$ is given by

$$\overrightarrow{\mathbf{x}}_{k+1} = \overrightarrow{\mathbf{\Phi}}_k \hat{\mathbf{x}}_k. \tag{C.3}$$

To find the covariance matrix of the predicted estimate, the above expression is expanded as

$$\begin{aligned}
\overrightarrow{\mathbf{x}}_{k+1} &= \overrightarrow{\mathbf{\Phi}}_k \hat{\mathbf{x}}_k \\
&= \overrightarrow{\mathbf{\Phi}}_k (\mathbf{x}_k + \epsilon_k) \\
&= \overrightarrow{\mathbf{\Phi}}_k \mathbf{x}_k + \overrightarrow{\mathbf{\Phi}}_k \epsilon_k
\end{aligned}$$

Solving Equation (C.1) for $\overrightarrow{\mathbf{\Phi}}_k \mathbf{x}_k$, and substituting this into the above equation produces

$$\overrightarrow{\mathbf{x}}_{k+1} = \mathbf{x}_{k+1} - \mathbf{G}_k \mathbf{U}_k + \overrightarrow{\mathbf{\Phi}}_k \epsilon_k \tag{C.4}$$

The covariance matrix may be found from this expression.

The expression for the covariance matrix of the prediction is given in the next section in the set of five Kalman Filter equations. In the Backward-Forward Extended Kalman Filter, the only equations that differ are the prediction equations developed in this section. While the prediction equations change in the Backward-Forward case, these results are still relevant.

C.3 Kalman Filter Equations

The Kalman Filter is described performed by the following set of five equations.

$$\overrightarrow{\mathbf{x}}_{k+1} = \overrightarrow{\mathbf{\Phi}}_k \hat{\mathbf{x}}_k \tag{C.5}$$

$$\overrightarrow{\mathbf{\Sigma}}_{k+1} = \overrightarrow{\mathbf{\Phi}}_k \mathbf{\Sigma}_{k+1} \overrightarrow{\mathbf{\Phi}}_k^\top + \mathbf{G}_k \mathbf{\Sigma}_{\mathbf{\Phi}} \mathbf{G}_k^\top \tag{C.6}$$

$$K_{k+1} = \overrightarrow{\mathbf{\Sigma}}_{k+1} \mathbf{M}_{k+1}^\top \left[\mathbf{M}_{k+1} \overrightarrow{\mathbf{\Sigma}}_{k+1} \mathbf{M}_{k+1} + \mathbf{\Sigma}_{\mathbf{\Omega}} \right]^{-1} \tag{C.7}$$

$$\hat{\mathbf{x}}_{k+1} = \overrightarrow{\mathbf{x}}_{k+1} - K_{k+1} \left(\mathbf{M}_{k+1} \overrightarrow{\mathbf{x}}_{k+1} - \mathbf{\Omega}_{k+1} \right) \tag{C.8}$$

$$\mathbf{\Sigma}_{k+1} = \overrightarrow{\mathbf{\Sigma}}_{k+1} - K_{k+1} \mathbf{M}_{k+1} \overrightarrow{\mathbf{\Sigma}}_{k+1} \tag{C.9}$$

These equations are given in [22]. These equations will be used and modified for the Backward-Forward Extended Kalman Filter. This section describes how to use prediction to improve the estimate of the subject's state (the geolocation estimate) and to improve the confidence in this estimate. In the next section, this idea will be extended to incorporate the subject's future states retrodicted to the current state being estimated.

C.4 Adding Retrodiction

Rather than only predicting forward to the subject's next state, the Forward-Backward Extended Kalman Filter also retrodicts to the subject's previous state. Therefore, the prediction and retrodiction must be used together to provide a better estimate of the subject's state and covariance matrix.

The best estimate of the subject's state at $k-1$ and at $k+1$ are described by

$$\hat{\mathbf{x}}_{k-1} \sim \mathcal{N}\left(\mathbf{x}_{k-1}, \mathbf{\Sigma}_{X_{k-1}}\right) \tag{C.10}$$

$$\hat{\mathbf{x}}_{k+1} \sim \mathcal{N}\left(\mathbf{x}_{k+1}, \mathbf{\Sigma}_{X_{k+1}}\right) \tag{C.11}$$

From Equation C.5, it can be can seen that the best prediction and retrodiction to the subject's state at k is given as

$$\overrightarrow{\mathbf{x}}_k = \overrightarrow{\mathbf{\Phi}}_{k-1}\hat{\mathbf{x}}_{k-1} \tag{C.12}$$

$$\overleftarrow{\mathbf{x}}_k = \overleftarrow{\mathbf{\Phi}}_{k+1}\hat{\mathbf{x}}_{k+1} \tag{C.13}$$

Now he ideal means of combining the prediction and retrodiction must be found to provide the best estimate of $\mathbf{x}_k$ using the prior and posterior subject states. This combined estimate is called $\overleftrightarrow{\mathbf{x}}_k$. The ideal combined estimate is the subject state at k that makes the prediction and retrodiction most likely. This becomes a likelihood estimation problem [18]. Therefore $\overleftrightarrow{\mathbf{x}}_k$ may be found as given by

$$\overleftrightarrow{\mathbf{x}}_k = \arg\max_{\mathbf{x}_k} \mathcal{L}\left(\overrightarrow{\mathbf{x}}_{k-1}, \overleftarrow{\mathbf{x}}_{k+1} | \mathbf{x}_k\right) \tag{C.14}$$

Expanding this equation requires knowledge of the distribution of $\overrightarrow{\mathbf{x}}_{k-1}$ and $\overrightarrow{\mathbf{x}}_{k+1}$. These distributions are Gaussian with covariances which may be calculated as shown in Equation (C.6). Positive constants are irrelevant to the arg max problem, so they are ignored in

$$\overleftrightarrow{\mathbf{x}}_k = \arg\max_{\mathbf{x}_k} e^{\left(-\frac{1}{2}\left(\overrightarrow{\mathbf{x}}_{k-1}-\mathbf{x}_k\right)^\top \overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1}\left(\overrightarrow{\mathbf{x}}_{k-1}-\mathbf{x}_k\right)\right)} e^{\left(-\frac{1}{2}\left(\overleftarrow{\mathbf{x}}_{k+1}-\mathbf{x}_k\right)^\top \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1}\left(\overleftarrow{\mathbf{x}}_{k+1}-\mathbf{x}_k\right)\right)} \tag{C.15}$$

The arg max is unaffected by taking the natural log of the argument. Therefore, by taking the natural log of this argument and eliminating the factor of $-1/2$, the equation is simplified to

$$\overleftrightarrow{\mathbf{x}}_k = \arg\min_{\mathbf{x}_k} \left(\overrightarrow{\mathbf{x}}_{k-1} - \mathbf{x}_k\right)^\top \overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} \left(\overrightarrow{\mathbf{x}}_{k-1} - \mathbf{x}_k\right) + \left(\overleftarrow{\mathbf{x}}_{k+1} - \mathbf{x}_k\right)^\top \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \left(\overleftarrow{\mathbf{x}}_{k+1} - \mathbf{x}_k\right) \tag{C.16}$$

This equation may now by solved for $\overleftrightarrow{\mathbf{x}}_k$ by finding the value of $\mathbf{x}_k$ such that the first derivative of the argument is zero. Calculating the derivative requires

$$\nabla_{\mathbf{x}} \left(\mathbf{x}^\top \mathbf{M} \mathbf{x} \right) = \left(\mathbf{M}^\top + \mathbf{M} \right) \mathbf{x} \tag{C.17}$$

Therefore, by using Equation C.17 on Equation C.16 and setting the result to zero, yielding

$$\left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1^\top} + \overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} \right) \left(\overrightarrow{\mathbf{x}}_{x-1} - \mathbf{x}_k \right) + \left(\overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1^\top} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \right) \left(\overleftarrow{\mathbf{x}}_{x+1} - \mathbf{x}_k \right) = 0 \tag{C.18}$$

Because covariance matrices are symmetric by their definition, this result may be further simplified to

$$\begin{aligned}
2\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} \left(\overrightarrow{\mathbf{x}}_{x-1} - \mathbf{x}_k \right) + 2\overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \left(\overleftarrow{\mathbf{x}}_{x+1} - \mathbf{x}_k \right) &= 0 \\
\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} \overrightarrow{\mathbf{x}}_{x-1} - \overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} \mathbf{x}_k + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \overleftarrow{\mathbf{x}}_{x+1} - \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \mathbf{x}_k &= 0 \\
\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} \mathbf{x}_k + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \mathbf{x}_k &= \overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} \overrightarrow{\mathbf{x}}_{x-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \overleftarrow{\mathbf{x}}_{x+1} \\
\left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \right) \mathbf{x}_k &= \overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} \overrightarrow{\mathbf{x}}_{x-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \overleftarrow{\mathbf{x}}_{x+1} \\
\mathbf{x}_k &= \left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \right)^{-1} \left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} \overrightarrow{\mathbf{x}}_{x-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \overleftarrow{\mathbf{x}}_{x+1} \right) \\
\overleftrightarrow{\mathbf{x}}_k &= \left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \right)^{-1} \left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} \overrightarrow{\mathbf{x}}_{x-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \overleftarrow{\mathbf{x}}_{x+1} \right)
\end{aligned} \tag{C.19}$$

The last step follows from Equation C.14. This equation provides a means of calculating the optimal prediction of $\mathbf{x}_k$. Using this result, the covariance matrix of the combined prediction $\overleftrightarrow{\mathbf{\Sigma}}_k$ can be calculated. The definition of the covariance matrix is used to find $\overleftrightarrow{\mathbf{\Sigma}}_k$ as given by

$$\overleftrightarrow{\mathbf{\Sigma}}_k = E\left[\left(\overleftrightarrow{\mathbf{x}}_k - E\left[\overleftrightarrow{\mathbf{x}}_k \right] \right) \left(\overleftrightarrow{\mathbf{x}}_k - E\left[\overleftrightarrow{\mathbf{x}}_k \right] \right)^\top \right] \tag{C.20}$$

The $\overleftrightarrow{\mathbf{x}}_k - E\left[\overleftrightarrow{\mathbf{x}}_k\right]$ term will be tackled first. It will be expanded and errors in the state estimate at $k-1$ and $k+1$ will be taken in consideration as in Section C.2. Before examining this term, $\overleftrightarrow{\mathbf{x}}_k$ is expanded out as follows via Equation C.5 and Equation C.19 as given by

$$\begin{aligned} &\text{Let } \mathbf{A} = \left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1}\right)^{-1} \\ &\overleftrightarrow{\mathbf{x}}_k = \mathbf{A}\left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1}\overrightarrow{\mathbf{\Phi}}_{k-1}\hat{\mathbf{x}}_{k-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1}\overleftarrow{\mathbf{\Phi}}_{k+1}\hat{\mathbf{x}}_{k+1}\right) \end{aligned} \tag{C.21}$$

From Equations (C.3) and (C.4), can be expanded $\overrightarrow{\mathbf{\Phi}}_{k-1}\hat{\mathbf{x}}_{k-1}$ and $\overleftarrow{\mathbf{\Phi}}_{k+1}\hat{\mathbf{x}}_{k+1}$ as

$$\begin{aligned} \overleftrightarrow{\mathbf{x}}_k &= \mathbf{A}\left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1}\left(\mathbf{x}_k - \mathbf{G}_{k-1}\mathbf{U}_{k-1} + \overrightarrow{\mathbf{\Phi}}_{k-1}\epsilon_{k-1}\right) + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1}\left(\mathbf{x}_k - \mathbf{G}_{k+1}\mathbf{U}_{k+1} + \overleftarrow{\mathbf{\Phi}}_{k+1}\epsilon_{k+1}\right)\right) \\ &= \mathbf{A}\left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1}\mathbf{x}_k + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1}\mathbf{x}_k\right) + \mathbf{A}\left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1}\left(\overrightarrow{\mathbf{\Phi}}_{k-1}\epsilon_{k-1} - \mathbf{G}_{k-1}\mathbf{U}_{k-1}\right) + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1}\left(\overleftarrow{\mathbf{\Phi}}_{k+1}\epsilon_{k+1} - \mathbf{G}_{k+1}\mathbf{U}_{k+1}\right)\right) \end{aligned} \tag{C.22}$$

The left-hand term in the final equation is constant since $\mathbf{x}_k$ is the true value of $\mathbf{x}$ at k. The right-hand term mean 0 because ϵ_{k-1} and $\epsilon_k + 1$ are mean-zero. Recall that it has been shown in Equation (2.53) that the NLO is an unbiased estimator. Therefore, in the $\overleftrightarrow{\mathbf{x}}_k - E\left[\overleftrightarrow{\mathbf{x}}_k\right]$ term, the constant factors are subtracted out since the expected value of a constants yields the same constant. The expected value of the right hand term is zero. Therefore,

$$\overleftrightarrow{\mathbf{x}}_k - E\left[\overleftrightarrow{\mathbf{x}}_k\right] = \mathbf{A}\left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1}\left(\overrightarrow{\mathbf{\Phi}}_{k-1}\epsilon_{k-1} - \mathbf{G}_{k-1}\mathbf{U}_{k-1}\right) + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1}\left(\overleftarrow{\mathbf{\Phi}}_{k+1}\epsilon_{k+1} - \mathbf{G}_{k+1}\mathbf{U}_{k+1}\right)\right). \tag{C.23}$$

Substituting this result into Equation (C.20) produces an equation too large to write out here. Therefore, the result of this substitution will be shown by placing the $\mathbf{U}_{k-1}$ and $\mathbf{U}_{k+1}$ terms into a variable $\mathcal{U}$. The expansion of the substitution shown below is used to

demonstrate that each term in Equation (C.23) may be analyzed separately, ie

$$\begin{aligned}
\overleftrightarrow{\mathbf{\Sigma}}_k &= E\left[\left(\mathbf{A}\left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1}\overrightarrow{\mathbf{\Phi}}_{k-1}\epsilon_{k-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1}\overleftarrow{\mathbf{\Phi}}_{k+1}\epsilon_{k+1} + \mathcal{U}\right)\right)\left(\mathbf{A}\left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1}\overrightarrow{\mathbf{\Phi}}_{k-1}\epsilon_{k-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1}\overleftarrow{\mathbf{\Phi}}_{k+1}\epsilon_{k+1} + \mathcal{U}\right)\right)^{\top}\right] \\
\overleftrightarrow{\mathbf{\Sigma}}_k &= E\left[\left(\mathbf{A}\left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1}\overrightarrow{\mathbf{\Phi}}_{k-1}\epsilon_{k-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1}\overleftarrow{\mathbf{\Phi}}_{k+1}\epsilon_{k+1} + \mathcal{U}\right)\right)\left(\left(\epsilon_{k-1}^{\top}\overrightarrow{\mathbf{\Phi}}_{k-1}^{\top}\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} + \epsilon_{k+1}^{\top}\overleftarrow{\mathbf{\Phi}}_{k+1}^{\top}\overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} + \mathcal{U}^{\top}\right)A^{\top}\right)\right] \\
\overleftrightarrow{\mathbf{\Sigma}}_k &= \mathbf{A}E\left[\left(\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1}\overrightarrow{\mathbf{\Phi}}_{k-1}\epsilon_{k-1} + \overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1}\overleftarrow{\mathbf{\Phi}}_{k+1}\epsilon_{k+1} + \mathcal{U}\right)\left(\epsilon_{k-1}^{\top}\overrightarrow{\mathbf{\Phi}}_{k-1}^{\top}\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1} + \epsilon_{k+1}^{\top}\overleftarrow{\mathbf{\Phi}}_{k+1}^{\top}\overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} + \mathcal{U}^{\top}\right)\right]A^{\top}
\end{aligned} \tag{C.24}$$

Due to the linearity of expected value [7], the expected values of the cross-terms may be examined separately. Since the only random variables present are ϵ_{k-1} and ϵ_{k+1}, the expected value will only operate on these. One of the cross-terms is

$$\overrightarrow{\mathbf{\Sigma}}_{k-1}^{-1}\overrightarrow{\mathbf{\Phi}}_{k-1}E\left[\epsilon_{k-1}\epsilon_{k+1}^{\top}\right]\overleftarrow{\mathbf{\Phi}}_{k+1}^{\top}\overleftarrow{\mathbf{\Sigma}}_{k+1}^{-1} \tag{C.25}$$

Note that from [7]. The covariance matrix of two random vectors X and Y may be described as

$$\Sigma(X, Y) = E\left[XY^{\top}\right] - E\left[X\right]E\left[Y\right]^{\top} \tag{C.26}$$

If X and Y are mean-zero, then this simplifies to

$$\Sigma(X, Y) = E\left[XY^{\top}\right] \tag{C.27}$$

since the subject's state estimates are mean-zero and are based on independent identically distributed measurements, their covariance is zero. This same argument holds for the expected value of the $\mathbf{U}$ and ϵ cross-terms. Therefore, Equation (C.24) simplifies to the expected value of the like-terms as shown below.

$$
\begin{aligned}
\overleftrightarrow{\boldsymbol{\Sigma}}_k &= \mathbf{A}E\left[\overrightarrow{\boldsymbol{\Sigma}}_{k-1}^{-1}\overrightarrow{\boldsymbol{\Phi}}_{k-1}\epsilon_{k-1}\epsilon_{k-1}^{\top}\overrightarrow{\boldsymbol{\Phi}}_{k-1}^{\top}\overrightarrow{\boldsymbol{\Sigma}}_{k-1}^{-1} + \overleftarrow{\boldsymbol{\Sigma}}_{k+1}^{-1}\overleftarrow{\boldsymbol{\Phi}}_{k+1}\epsilon_{k+1}\epsilon_{k+1}^{\top}\overleftarrow{\boldsymbol{\Phi}}_{k+1}^{\top}\overleftarrow{\boldsymbol{\Sigma}}_{k+1}^{-1} + \mathcal{U}\mathcal{U}^{\top}\right]A^{\top} \\
&= \mathbf{A}\left(\overrightarrow{\boldsymbol{\Sigma}}_{k-1}^{-1}\overrightarrow{\boldsymbol{\Phi}}_{k-1}E\left[\epsilon_{k-1}\epsilon_{k-1}^{\top}\right]\overrightarrow{\boldsymbol{\Phi}}_{k-1}^{\top}\overrightarrow{\boldsymbol{\Sigma}}_{k-1}^{-1} + \overleftarrow{\boldsymbol{\Sigma}}_{k+1}^{-1}\overleftarrow{\boldsymbol{\Phi}}_{k+1}E\left[\epsilon_{k+1}\epsilon_{k+1}^{\top}\right]\overleftarrow{\boldsymbol{\Phi}}_{k+1}^{\top}\overleftarrow{\boldsymbol{\Sigma}}_{k+1}^{-1} + \mathcal{U}\mathcal{U}^{\top}\right)A^{\top} \\
&= \mathbf{A}\left(\overrightarrow{\boldsymbol{\Sigma}}_{k-1}^{-1}\overrightarrow{\boldsymbol{\Phi}}_{k-1}\boldsymbol{\Sigma}_{\mathbf{x}_{k-1}}\overrightarrow{\boldsymbol{\Phi}}_{k-1}^{\top}\overrightarrow{\boldsymbol{\Sigma}}_{k-1}^{-1} + \overleftarrow{\boldsymbol{\Sigma}}_{k+1}^{-1}\overleftarrow{\boldsymbol{\Phi}}_{k+1}\boldsymbol{\Sigma}_{\mathbf{x}_{k+1}}\overleftarrow{\boldsymbol{\Phi}}_{k+1}^{\top}\overleftarrow{\boldsymbol{\Sigma}}_{k+1}^{-1} + \mathcal{U}\mathcal{U}^{\top}\right)A^{\top} \qquad \text{(C.28)}
\end{aligned}
$$

Likewise, expanding out and simplifying the $\mathcal{U}\mathcal{U}^{\top}$ term produces

$$
\mathcal{U}\mathcal{U}^{\top} = \overrightarrow{\boldsymbol{\Sigma}}_{k-1}^{-1}\mathbf{G}_{k-1}\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}\mathbf{G}_{k-1}^{\top}\overrightarrow{\boldsymbol{\Sigma}}_{k-1}^{-1} + \overleftarrow{\boldsymbol{\Sigma}}_{k+1}^{-1}\mathbf{G}_{k-1}\boldsymbol{\Sigma}_{\boldsymbol{\Phi}}\mathbf{G}_{k+1}^{\top}\overleftarrow{\boldsymbol{\Sigma}}_{k+1}^{-1} \qquad \text{(C.29)}
$$

We now have a means of calculating the covariance matrix of the combined prediction and retrodiction estimate of the subject's state. Equations C.28 and C.19 may be used in the Kalman Filter equations given in Section C.3.

In this way, the subject's track is taken advantage of to refine the state estimates and the confidences in those estimates. The confidence in the transition matrix may not be known. Therefore, it is common to use a guess at the first iteration of the Kalman Filter.

This approach was deemed inappropriate since the same LOS measurements may be used to geolocate the subject at different times. Therefore, the geolocation estimates are not linearly independent thus violating a Kalman Filter assumption.

Bibliography

[1] Bar-Shalom, Y., X.R. Li, T. Kirubarajan, and J. Wiley. *Estimation with applications to tracking and navigation*. Wiley Online Library, 2001.

[2] Bates, Douglas M and Donald G Watts. *Nonlinear regression: iterative estimation and linear approximations*. Wiley Online Library, 1988.

[3] Bierman, Gerald J. *Factorization methods for discrete sequential estimation*. Courier Dover Publications, 2006.

[4] Bishop, A.N., B. Fidan, B. D O Anderson, P.N. Pathirana, and K. Dogancay. "Optimality Analysis of Sensor-Target Geometries in Passive Localization: Part 2 - Time-of-Arrival Based Localization". *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, 13–18. 2007.

[5] Burchett, L., S. Hartzell, G. Hoffar, J. Mautz, C. Taylor, and A. Terzuoli. "Angle of arrival geolocation using non-linear optimization". *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, 1–4. 2012. ISSN 2153-6996.

[6] Call, Brandon, Randy Beard, Clark Taylor, and Blake Barber. "Obstacle avoidance for unmanned air vehicles using image feature tracking". *AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO*. 2006.

[7] Casella, George and Roger L Berger. *Statistical inference*, volume 70. Duxbury Press Belmont, CA, 1990.

[8] Chatterjee, Samprit and Ali S Hadi. *Regression analysis by example*, volume 607. Wiley-Interscience, 2006.

[9] Chehri, Abdellah, Paul Fortier, and Pierre-Martin Tardif. "Geo-location with wireless sensor networks using non-linear optimization". *Proceedings of International Journal of Computer Science and Network Security (IJCSNS)*, 145–154, 2008.

[10] Chitte, S.D., S. Dasgupta, and Zhi Ding. "Source Localization from Received Signal Strength Under Log-Normal Shadowing: Bias and Variance". *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, 1–5. 2009.

[11] Dogandzic, A., J. Riba, G. Seco, and A.L. Swindlehurst. "Positioning and navigation with applications to communications [from the Guest Editors]". *Signal Processing Magazine, IEEE*, 22(4):10–11, 2005. ISSN 1053-5888.

[12] Fishman, George S. *Monte Carlo: concepts, algorithms, and applications*, volume 1196. Springer New York, 1996.

[13] Flury, Bernard. *A first course in multivariate statistics*. Springer, 1997.

[14] Gregory, Micah D, Zikri Bayraktar, and Douglas H Werner. "Fast optimization of electromagnetic design problems using the covariance matrix adaptation evolutionary strategy". *Antennas and Propagation, IEEE Transactions on*, 59(4):1275–1285, 2011.

[15] Hartley, Richard and Andrew Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000.

[16] Hegarty, C.J. and E. Chatre. "Evolution of the Global Navigation SatelliteSystem (GNSS)". *Proceedings of the IEEE*, 96(12):1902–1917, 2008. ISSN 0018-9219.

[17] Hilland, D.H., G.S. Phipps, C.M. Jingle, and G. Newton. "Satellite threat warning and attack reporting". *Aerospace Conference, 1998 IEEE*, volume 2, 207–217 vol.2. 1998. ISSN 1095-323X.

[18] Kay, Steven M. *Fundamentals of statistical signal processing, volume I: Estimation theory (v. 1)*. Prentice Hall, 1993.

[19] Krooneberg, Pieter M. *Three-mode principal component analysis: Theory and applications*, volume 2. DSWO press, 1983.

[20] Kulaib, A. R., R.M. Shubair, M.A. Al-Qutayri, and J.W.P. Ng. "An overview of localization techniques for Wireless Sensor Networks". *Innovations in Information Technology (IIT), 2011 International Conference on*, 167–172. 2011.

[21] Larson, Ron and Bruce H Edwards. *Calculus*. Brooks/Cole, 2009.

[22] Leondes, Cornelius T. *Theory and applications of Kalman filtering*. Technical report, DTIC Document, 1970.

[23] Lowell, J.R. "Military applications of localization, tracking, and targeting". *Wireless Communications, IEEE*, 18(2):60–65, 2011. ISSN 1536-1284.

[24] Luo, Jiecai, E. Walker, P. Bhattacharya, and X. Chen. "A new TDOA/FDOA-based recursive geolocation algorithm". *System Theory (SSST), 2010 42nd Southeastern Symposium on*, 208–212. 2010. ISSN 0094-2898.

[25] Marcus, Marvin and Henryk Minc. *A survey of matrix theory and matrix inequalities*, volume 14. Courier Dover Publications, 1992.

[26] Misra, P and P Enge. *Global Positioning System: Signals, Measurements and Performance*. Ganga-Jamuna Press, 2011.

[27] Musicki, D., R. Kaune, and W. Koch. "Mobile Emitter Geolocation and Tracking Using TDOA and FDOA Measurements". *Signal Processing, IEEE Transactions on*, 58(3):1863–1874, 2010. ISSN 1053-587X.

[28] Robert, Christian P and George Casella. *Monte Carlo statistical methods*, volume 319. Citeseer, 2004.

[29] Roxin, A., J. Gaber, M. Wack, and A. Nait-Sidi-Moh. "Survey of Wireless Geolocation Techniques". *Globecom Workshops, 2007 IEEE*, 1–9. 2007.

[30] Seber, GAF and CJ Wild. *Nonlinear regression. 1989*. John Wiley & Sons Inc.

[31] Strang, Gilbert. *Linear Algebra and Its Applications*. Thomson Books/Cole, 2006.

[32] Wang, Jun-Lin and Jian-Yun Zhang. "Target Detection with Non-Cooperative Illuminators". *Antennas, Propagation EM Theory, 2006. ISAPE '06. 7th International Symposium on*, 1–5. 2006.

[33] Xiu, Jihong, Pu Huang, Jun Li, Youyi Li, and Li Chen. "Line-of-sight Locating algorithm for staring imaging aerial remote sensor". *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, volume 2, 273–277. 2010.

[34] Xu, Enyang, Zhi Ding, and S. Dasgupta. "Source Localization in Wireless Sensor Networks From Signal Time-of-Arrival Measurements". *Signal Processing, IEEE Transactions on*, 59(6):2887–2897, 2011. ISSN 1053-587X.

Vita

Stephen Hartzell was homeschooled until he was 15 when he began coursework at Sinclair Community College. While at Sinclair he began to work as a research assistant under Dr. Terzuoli. He completed a two-year electrical engineering transfer program for Wright State University. After transferring to Wright State University, Stephen finished his Bachelor of Science in Electrical Engineering in 2012 summa cum laude. At this time he began working on his Masters of Science in Electrical Engineering at the Air Force Institute of Technology.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
19–06–2014	Master's Thesis	Jan 2013 – Jun 2014

4. TITLE AND SUBTITLE

Non-Linear Optimization Applied to Angle-of-Arrival Satellite-Based Geolocation

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Hartzell, Stephen D., Civilian Student, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-T-14-J-7

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

INTENTIONALLY LEFT BLANK

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Geolocation is a common application for satellite systems. This involves estimating an object's location (herein called the subject) based on noisy satellite data. Many geolocation methods exist; however, none are tailored specifically for the unique problems faced by satellite systems. Some satellites are so far from the subject being localized that by the time the satellite receives a signal from the subject it might have moved appreciably. Furthermore, some satellites or terrestial sensors may be much closer to the subject than others. Therefore, sensors may need to be weighted based upon their distance to the subject being localized. In addition, even if a subject can be localized, the confidence in this localization may be unknown. Non-linear optimization is proposed, implemented, and analyzed as a means of geolocating objects and providing confidence estimates from passive satellite line-of-sight data. Non-linear optimization requires an initial estimate. This estimate is provided by a triangulation method. The non-linear optimization then improves upon this estimate iteratively by finding estimates that are more likely to have produced the observed line-of-sight measurements. The covariance matrix of the geolocation parameters being estimated is naturally produced by the optimization which provides quantified confidence in the geolocation estimate. Simulations are developed to provide a means of evaluating the performance of the non-linear optimization algorithm. It was found that non-linear optimization can reduce the average error in geolocation estimates, provide improved estimation confidence, and accurately estimate its geolocation confidence for some subjects. The results from the theoretical development of the non-linear optimization algorithm and its simulated performance is quantified and discussed.

15. SUBJECT TERMS

Geolocation, Non-Linear Optimization, Triangulation, Line-of-Sight, Angle-of-Arrival

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Andrew J. Terzuoli (ENG)
U	U	U	UU	146	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636 x4717 Andrew.Terzuoli@afit.edu