NON-LINEAR OPTIMIZATION APPLIED TO ANGLE-OF-ARRIVAL SATELLITE BASED GEO-LOCALIZATION FOR BIASED AND TIME DRIFTING SENSORS

THESIS

2nd Lt. Daniel Levy, 2nd Lt, USAF

AFIT-ENG-MS-16-M-032

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-16-M-032

NON-LINEAR OPTIMIZATION APPLIED TO ANGLE-OF-ARRIVAL SATELLITE BASED GEO-LOCALIZATION FOR BIASED AND TIME DRIFTING SENSORS

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

2nd Lt. Daniel Levy, B.S.E.E.,B.S.C.E.

2nd Lt, USAF

March 2016

AFIT-ENG-MS-16-M-032

NON-LINEAR OPTIMIZATION APPLIED TO ANGLE-OF-ARRIVAL SATELLITE BASED GEO-LOCALIZATION FOR BIASED AND TIME DRIFTING SENSORS

THESIS

2nd Lt. Daniel Levy, B.S.E.E.,B.S.C.E.
2nd Lt, USAF

Committee Membership:

Dr. A. J. Terzuoli
Chair

Dr. R. M. Martin
Member

Dr. C. N. Taylor
Member

Dr. W. J. Carpenter
Member

AFIT-ENG-MS-16-M-032

Abstract

Multiple sensors are used in a variety of geolocation systems. Many use Time Difference of Arrival (TDOA) or Received Signal Strength (RSS) measurements to locate the most likely location of a signal. When an object does not emit a classical RF signal, Angle of Arrival (AOA) measurements become more feasible than TDOA or RSS measurements. AOA measurements can be created from any sensor platform with any sort of camera. When location and attitude knowledge of the sensor passive objects can be tracked. A Non-Linear Optimization (NLO) method for calculating the most likely estimate from AOA measurements has been created in previous work. This thesis, modifies that algorithm to automatically correct AOA measurement errors by estimating the inherent bias and time-drift in the Inertial Measurement Unit (IMU) of the AOA sensing platform. Two methods are created to correct the sensor bias. One method corrects the sensor bias in post processing while treating the previous NLO method as a module. The other method directly corrects the sensor bias within the NLO algorithm by incorporating the bias parameters as a state vector in the estimation process. These two methods are analyzed using various Monte-Carlo simulations to check the general performance of the two modifications in comparison to the original NLO algorithm. These methods appear to improve performance by $10 - 60\%$ depending on the data.

Table of Contents

Page

Page

Page

List of Figures

List of Tables

List of Symbols

Symbol		Page
ε	Error vector	26
$\mathbf{H}$	Time transition matrix	26
$\widehat{\theta}$	Kinematic model estimate	26
B	Bias element (radians)	41
D	Drift Element (radians/second)	41
$\mathbf{\Sigma}_X$	State covariance matrix	44
DI	Drift Instability (radians/second$^{1/2}$)	46
μ	Standard mean value	47
σ	Scalar standard deviation value	47

NON-LINEAR OPTIMIZATION APPLIED TO ANGLE-OF-ARRIVAL SATELLITE BASED GEO-LOCALIZATION FOR BIASED AND TIME DRIFTING SENSORS

I. Introduction

1.1 Introduction

Object localization and tracking is a growing field as more sophisticated space based sensors are developed. In the field of localization, geolocalization is the process of estimating an object position in relation to the world. Global Navigation Satellite System (GNSS) is a well-known and successful Time of Arrival (TOA) localization technique using multiple transmitters and one receiver. Transmitting objects can also be geolocated with multiple receivers in a very similar fashion. Given a close range configuration of sensors, Received Signal Strength (RSS) can be used to successfully geolocate an active object. However, for long range geolocalization and tracking of a passive object, AOA geolocalization is a much more useful method. Any type of two dimensional sensor can make an Angle-of-Arrival (AOA) measurement. An AOA, location and attitude measurement can be used to calculate a LOS vector measurement which can be used to geolocate and track an object. AOA geolocalization can also be used in optical navigation. Of many different optical navigation techniques, the Simultaneous Localization and Mapping (SLAM) problem is a commonly growing secondary navigation method in a GNSS denied environment, which uses AOA measurements to geolocate objects. AOA can be very useful when tracking passive objects as well, such as meteors, deactivated satellites, and other space debris. AOA geolocalization normally uses

a triangulation approach which works well for stationary objects. This thesis will discuss AOA object tracking and geolocalization using a Non-Linear Optimization (NLO) method as a way to find the most likely path of an object given Line of Sight (LOS) measurements from multiple sensors. This method will be tested on different simulated path types and sensing scenarios. It will use realistic sensor models as well. The AOA geolocalization algorithm developed in this thesis will allow real sensors LOS measurements to accurately track an object.

1.2 Background

This section contains important background information for the NLO AOA to include, localization, coordinate systems, sensors, sensor configurations and path optimization techniques. These topics will cover the basic ideas governing the algorithm development in this thesis.

Localization.

Localization is the process of estimating the position of an object in a specific coordinate system. A good localization estimate produces an accurate position of an object and in the case of multiple measurements over time, the kinematic model of the object. Given sufficient knowledge about the sensor configuration and variance, the confidence of a localization can also be calculated which can be a useful real time metric of the localization algorithm performance without apriori knowledge. Many different factors can affect the accuracy of a position estimate. These factors can include amount and statistical model of error in the LOS measurements, geometry of the sensor configuration, accuracy of the object trajectory model or the quality of the algorithm used to estimate the path.

In any localization algorithm, it is important to use a relevant, consistent and simple coordinate system. Because this algorithm is intended for geolocalization, the ECEF coordinate system is used. ECEF is an x-y-z Cartesian coordinate system. The x-axis is a vector that passes through the Greenwich meridian, while the z-axis travels through the North Pole along the Earth's rotational axis. The y-axis vector passes through the equator orthogonal to the x-axis and z-axis to create a right-handed three dimensional orthogonal basis.

The three most common localization methods are RSS, TOA and AOA. In RSS and TOA localization, a range estimate is usually made between the object and the configuration of sensors. From these range estimates, a position in relation to the sensors can be calculated. AOA localization starts with a direction measurement instead of a range measurement. From two dimensional sensors, LOSs are observed between an object and multiple sensors. These LOSs can be described by the position of the sensor and two angles forming a vector traveling through 3-space. From the LOS data, a point can be estimated. Traditionally, a triangulated point estimate will be the point in 3-space with the least Euclidean distance between the three dimensional LOS vectors. However, the NLO approach will incorporate the possible sensor error and create an estimate with the least amount of sensor angle error. This NLO algorithm will be described in detail later in chapters II and III of this thesis.

Sensors and Sensor Configuration.

The NLO algorithm in this thesis is applicable to any configuration of two dimensional LOS sensors. A LOS sensor could be a camera, IR sensor, beam forming systems, hyper spectral detection or anything that can produce a direction estimate between the sensor and a transmitting object. The simulations later described in this thesis

use simulated space-based sensors. These space based sensors are in Low Earth Orbit (LEO) with generic LOS sensors on board detecting the same object at non time incidental measurements.

Generally, stationary LOS sensors will contain Gaussian angle error which would be stationary as the sensor ages. However, a mobile LOS measurement system will use the IMU to calculate attitude which is necessary for the actual two dimensional AOA sensor to make a useful angle measurement. IMUs do not generally have a Wide Sense Stationary (WSS) error distribution. Instead, IMUs may contain a small bias in measurement. Because each IMU measurement cumulatively affects the attitude calculation, IMU error is best described by a random angle walk. Random angle walks may have a biased angle error in one direction or another. Because of the cumulative biased error, IMU error will drift a particular direction as time passes. The direction of the drift will depend on the IMU and the environment. For many space based IMUs, the error will drift for several minutes until the true Space vehicle (SV) orientation is re-calculated through other more accurate means and corrected. This is commonly done with star tracking. When an IMU drifts over time, the beginning of a sensing period may be initially biased. Each sensor system will contain an unknown bias at the start time of object detection that will add a constant error into the system as well as the IMU time drift and the Gaussian like jitter error from the generic WSS error in the other LOS measurement system elements.

In all forms of localization the geometric configuration of the sensors in relation to the tracked object plays a role in the error induced in the object localization. Propagating the error through the geolocation algorithm produces a three dimensional multivariate Gaussian like distribution of point estimates. Under certain sensor configurations and sensors with a low enough error variance, the error can be assumed to be Gaussian. These conditions depend on the error variance of the sensor and the angle of intersection

between the LOS vectors produced by the sensor systems. The specific conditions for this assumption will be shown later in this thesis as well as the methodology for finding the hard limit. The localization distribution variance grows larger and smaller depending on these conditions as well. As the LOS intersections approach zero and 180 degrees, the volume of point estimate confidence ellipsoids grows larger. Computing the Cramer-Rao Lower Bound (CRLB) of the point estimates shows that the smallest localization error occurs when the LOS vectors are orthogonal to each other.

Path Optimization.

The algorithm developed in this thesis optimizes path localization by finding the path with the least amount of sensor error given a set of LOS data. To find the optimum path, Newton's method is used. Newton's method begins with an initial guess and then iterates closer to the most likely path using a Jacobian built from the partial derivatives calculated from the change of ECEF position and the LOS angle error. The position is iteratively recalculated until the iterative position change is minimal. Given an adequate initial guess, a stable system and an existing error minimum, Newton's method will converge on an optimum solution.

Calculating the Jacobian for this NLO algorithm relies on the assumed path model. The path estimate is calculated based on stationary objects, moving objects with a constant velocity, and objects with a constant acceleration. The higher the order of the path model, the larger the Jacobian grows. For higher order path estimates to have an optimum solution, the iterative method must include an adequate amount of point measurements to detect those higher order elements. Otherwise, the algorithm will attempt optimize an under determined system which does not have a solution. An over determined estimate can be found in one of two ways in this application. Either a sliding

window of sufficient size will use a number of surrounding LOS measurements to create an acceleration-velocity-position object estimate, or all of the observed estimates can be used to determine the entire path. Using every single estimate is generally the most accurate way of optimizing over the entire path; however, it creates a very large Jacobian which can be computationally taxing and require an enormous amount of memory for each iterative step.

In chapter III a new trajectory segmentation algorithm is developed. This trajectory segmentation algorithm estimates windows to break up LOS measurements according to the NLO path model. If an object with a very complicated path is being observed, the acceleration of the object will not be constant for the entire path. The trajectory segmentation algorithm can break a complicated path into different time groups and window the LOS measurements according to those time groups. The NLO will then be applied to those discretely separated time groups.

1.3 Problem Statement

The algorithm in this thesis is developed to create the most accurate object track estimate from AOA measurements. The algorithm must have the capability to handle a variety of paths, sensing scenarios and work well for space based LOS sensors. The algorithm must handle the type of statistical error existing in space based LOS sensor systems. The AOA NLO has already been tested on simple simulated data but has yet to be fully developed for real data. The largest obstacle in this transition is detecting and adjusting for the IMU induced random walk bias and drift as well estimating more complicated paths that do not effectively fit the constant acceleration model.

Accurate error volumes are also necessary for this project to depict the confidence and performance of different sensor systems and estimates without a priori knowledge.

These error surfaces or error ellipsoids are developed by computing the Covariance Intersection (CI) of each sensor configuration which incorporates self correlating sensor measurements. The volume of these error ellipsoids depicts the space that the algorithm is confident the object exists in at a set statistic.

1.4 Methodology

The path segmentation algorithm focuses on breaking up an observed noisy path estimate into easily digestible segments for the NLO algorithm. The NLO algorithm bases its estimate on a constant acceleration or low-jerk trajectory model. Once this condition is violated, the NLO algorithm breaks down. The method used to segment the path will be described in detail in chapter III. Once the path segmentation method has broken up the path into lower-jerk segments, the NLO algorithm will be applied to each section.

The bias and drift estimation attempts to calculate out the non-Gaussian error in each sensor system. The bias and drift estimation method operates without information about the bias and drift of each sensor. Estimating a sensor bias without any prior knowledge is a challenging problem but it is possible given multiple sensors and measurements. The estimation method will take multiple sensors with independent biases and drifts and transform the effective error into a zero-mean Gaussian distribution. This can be done with methods which will be described in detail in chapter III.

In order to characterize the performance of the AOA NLO algorithms, Monte-Carlo simulations will test the accuracy of the calculated error of the system. By generating random tests, the effective error ellipsoids of the localization algorithm can be estimated and compared with the calculated error ellipsoids. As well as error surface validation, the Monte-Carlo simulation will test the effect of a variety of path types and sensing scenarios to include complicated paths with large jerk elements, paths with time gaps without

measurements or few measurements, paths with heavy sensor bias, drift of other error, varying sensor sampling rates, different number of sensors, different measurements per sensor and different geometric sensor configurations. These tests will characterize the performance and limits for each of the NLO algorithms developed in this thesis, those developed by Sprang and Hartzell, and triangulation as a performance comparison [1],[2].

1.5 Conclusion

The purpose of this thesis is to put the finishing touches on an already developed localization algorithm. Previously, the NLO algorithm has worked very well on simple simulated data as shown by Sprang and Harzell. The algorithm works well but has many areas of improvement. The AOA NLO algorithms have well working methods for estimating regular trajectories for non-correlated sensors and constant acceleration kinematic paths, but it has not been developed thoroughly for a realistic LOS sensor system error or complicated trajectory types. This thesis attempts to alter the AOA NLO algorithm so that sensor bias and drift is calculated out of the estimate. Once these adjustments are made, far more accurate geolocalization estimates are made on a variety of objects using a variety of object sensing scenarios [1],[2].

II. Background

2.1 Introduction

In this chapter, the theoretical background will be discussed in detail as well as the most up to date literature in the field. This chapter will include detailed information on geolocation algorithms, geolocation confidence metrics, LOS geometry, optimization techniques and the previous AOA NLO, trajectory modeling and estimation techniques, and satellite-based LOS error. This research is meant to improve the previously existing AOA NLO geolocation algorithms by accounting for realistic errors and more complicated object paths.

2.2 Line of Sight Geometry

In many geolocalization techniques, two dimensional sensors are used to create AOA measurements of an object. AOA can be measured with any sort of optical, infrared or direction estimating sensor. AOA estimates between an object and a sensor can be expressed by two angle measurements θ and ϕ expressed in

$$\begin{aligned} \theta =& tan^{-1}\left(\frac{\Delta y}{\Delta x}\right), \\ \phi =& tan^{-1}\left(\frac{\sqrt{\Delta x^2+\Delta y^2}}{\Delta z}\right). \end{aligned} \tag{1}$$

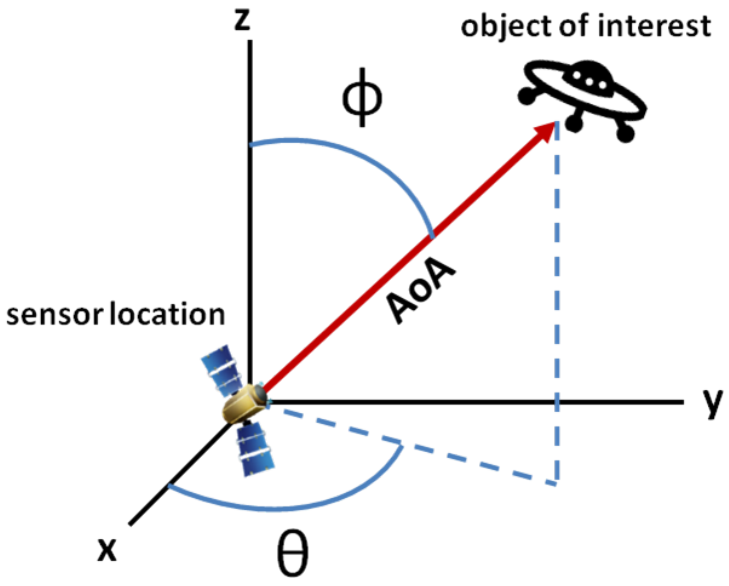

Figure 1. Angle of Arrival Measurement [1].

or in Figure 1 as a graphical representation where x, y, z are ECEF cartesian position coordinates. The algorithm developed in this thesis uses this base line measurement model for AOA sensors.

Triangulation.

There are many methods to gather information from AOA measurements. In [3], many methods are described to gather three dimensional information from multiple cameras. While [3] focuses on optical devices, many of the principles apply to all AOA sensors. The most common method for multi-view three dimensional estimation is triangulation, which [3] describes in detail. Triangulation finds the minimal Euclidean distance between two or more LOS vectors. A depiction of this can be seen in Figure 2.

While the triangulation method is efficient computationally, it does not account for the varying accuracy of each sensor and the geometric sensor configuration. Triangulation

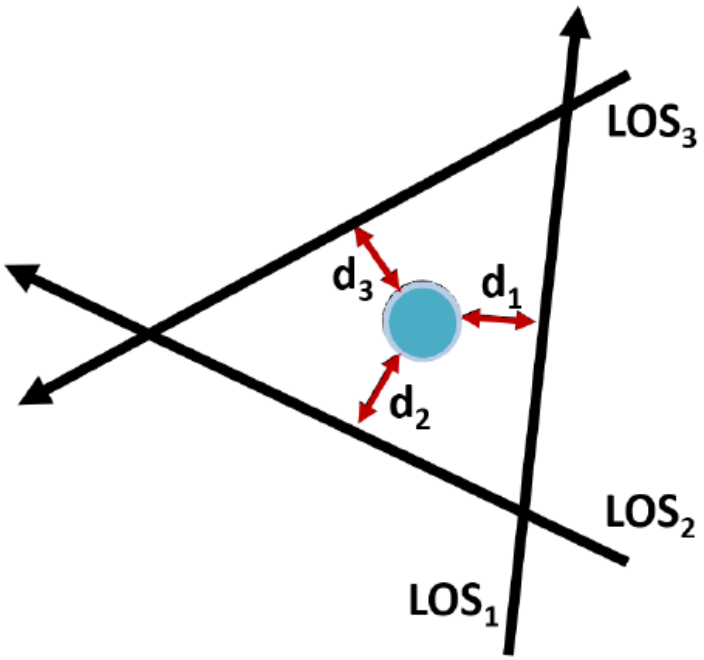

Figure 2. Triangulation Example [1].

would be an inaccurate method given a sensing scenario where one AOA sensor is closer or more accurate than the other, because it weighs each LOS vector equally. In this geolocalization scenario, the more accurate or closer an AOA sensor, the more weight that sensor should be given than the less trusted sensor. The NLO algorithm will use a Jacobian matrix similar to the one shown in (4) to account for AOA error instead of Euclidean LOS error and create better geolocalization estimates. According to [3], AOA triangulation may not be Gaussian; however, under conditions with sufficient angular separation between sensors and a small enough sensor error variance, it is a safe assumption. With an assumed Gaussian geolocalization error, the CRLB can be used to characterize the confidence of the system accurately as a confidence ellipsoid [3].

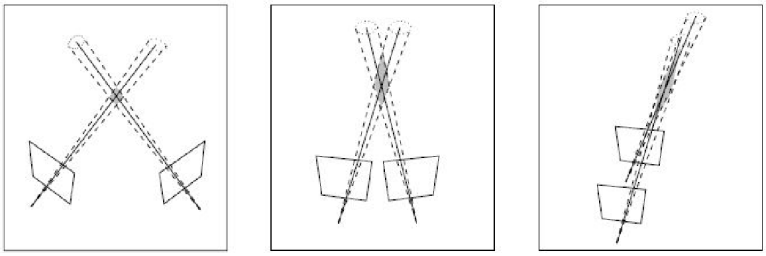

Figure 3. Reconstruction Uncertainty Regions (shaded area) for different Camera Positions [3, p. 435].

LOS Error Propagation.

Much will be discussed regarding sensor error later in this chapter. It is important to note how LOS sensor error will translate into locational error after triangulation or the NLO algorithm is performed. The use of CI in geolocalization problems as shown by [1],[4],[5] is an accurate way of calculating potential error propagation in an AOA geolocalization problem. The CI is an accurate estimate as the CRLB which describes how certain a position estimate can be [1],[4],[5]. As sensor variance or sensor geometry changes, the amount of uncertainty in a position estimate grows and diminishes. This relationship is reflected mathematically through the CRLB. [3] discusses a very similar problem in relation to the Probability Density Function (PDF) of 3D camera reconstruction. As seen in Figure 3, the shape of an uncertainty region is dependent on the intersection of the two sensors LOS measurements. The more parallel the LOS vectors become, the larger and more uncertain the 3D reconstruction error becomes [3].

[3] continues to investigate how a Gaussian like pixel error in cameras translates into 3D reconstruction error. Figure 4 shows an example of that transformation. In this example two cameras are simulated looking at an object a set distance away with an ideal LOS intersection of 90 degrees. The example in Figure 4 is simulated with a unusually high amount of variance in Gaussian pixel error. While this particular example is not even close to a realistic amount of error, it shows that even the most ideal sensor configuration

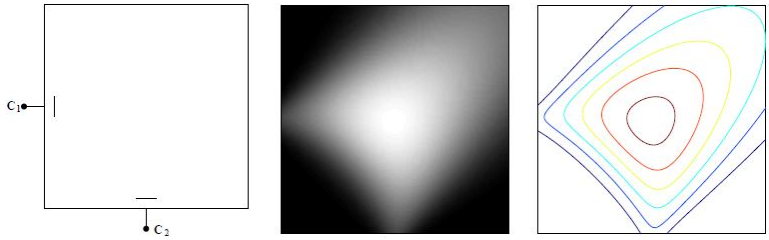

Figure 4. Triangulation PDF given large variance and 90 degree LOS intersection; camera configuration left, heat map center and contour plot right [3] pg 322.

can create a non-Gaussian localization estimate distribution. This thesis will not strictly focus on modeling camera error like [3] does; however, the problem can translate to any LOS sensing system [3]. CI is a useful way of correlating measurements from one sensor with themselves with statistically independent self correlating sensor measurements to create a confidence ellipse. Translating the CI into a confidence ellipse assumes a Gaussian type loacalization error. It is crucial to strictly define where Gaussian measurements transform into Gaussian like localization estimates for this reason. The Royston test is an easy method to test whether or not a data set is Gaussian or not developed in [6]. An approach to defining the limit where AOA measurements are Gaussian using the Royston test will be discussed in detail in Chapter III.

IMU Error Propagation.

In [7] IMU error is discussed in heavy detail. IMUs measure attitude through gyroscopes. There are three commonly used gyroscopes, Mechanical, Fiber Optic Gyroscope (FOG), and Micro-machined Electromechanical Systems (MEMS). A mechanical gyroscope contains a very stable spinning wheel mounted on two gimbals. Mechanical gyroscopes are rarely used due to all of the moving parts involved which allows friction to heavily affect the accuracy. FOG gyroscopes measure the speed light travels through a fiber optic cable and translates the speed into angular velocity. While

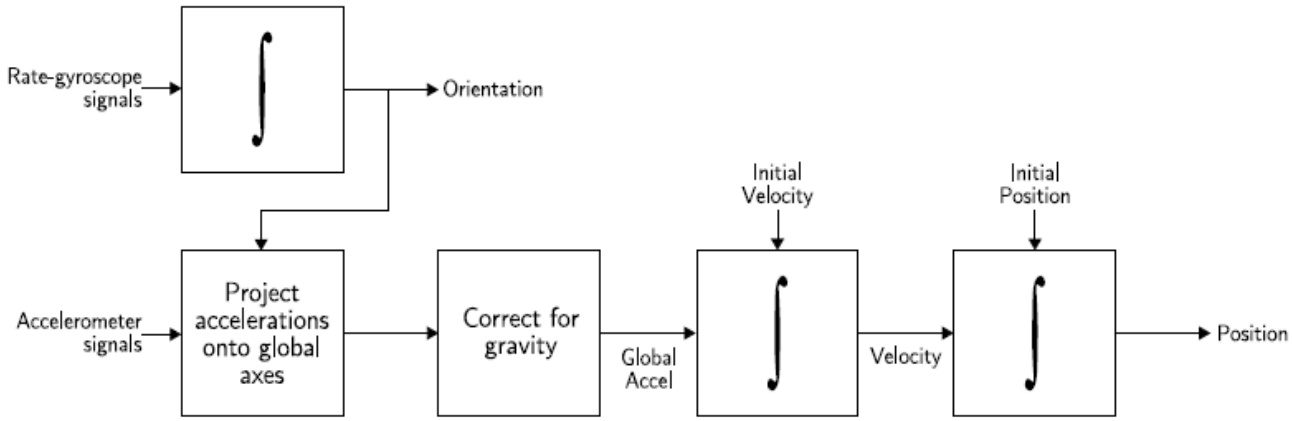

Figure 5. Inertial Navigation Algorithm [7].

FOGs contain no moving parts, the accuracy is dependent on how large the fiber optic coils are. Finally MEMS gyroscopes use micro-mechanical devices to measure the Coriolis effect which can translate into angular rotation. Even though MEMS gyroscopes could not be as accurate as FOG gyroscopes, MEMS gyroscopes are small, cheap to produce in bulk and rugged. Because of these many benefits, MEMSs are widely used and [7] focuses his report on the performance of these gyroscopes. In [7], characterizing IMU error revolved around navigation and his IMU navigation algorithm can be seen in Figure 5. For navigation every translated element from both a gyroscope and an accelerometer are important. Orientation is derived from the gyroscope signals integrated over time, while the velocity and acceleration are derived with integrations from both the accelerometer and the gyroscope [7].

MEMS gyroscopes tend to contain a tiny bias in measurement. This bias can be measured by averaging the angle measurement when the gyroscope is stationary. This bias propagates linearly in time when translated to IMU orientation. While measuring the IMU bias could be used to correct future IMU measurements, it will not be a static bias. MEMS measurement biases are affected by fluctuations in temperature in a nonlinear way. These temperature fluctuations can be caused by the environment or from MEMS self heating. In

addition to the gyroscopic bias, thermo-mechanical white noise adds a zero mean Jitter to angular measurements as well. The error model used in this thesis will be a random walk where each random walk step is a random Gaussian value averaging at some unknown gyroscopic bias [7].

The time drifting random walk is a similar model used in [8] which analyzes aeronautical navigation methods during a hypothetical interruption of Global Positioning System (GPS) service. [8] defines a gyroscope error function as

$$b(t) = b_0 + b_1(t) + b_w \tag{2}$$

where $b(t)$ is a biased sensor error function dependent on time, b_0 is the null shift, $b_1(t)$ is the drift function of the gyroscope and b_w is the instantaneous noise of the gyroscope. [8] attempts to correct for the gyroscope bias by finding the Alan variance of some FOG gyroscopes. [8] found that FOG gyroscopes are not significantly affected by temperature and have very low bias instability. With a low bias instability, a gyroscope's bias can be estimated and corrected for before integration in a navigation or object tracking system [8].

This analysis was done on a system that was not exposed to very large temperature changes that may be experienced by a space based system, as well as only for FOG gyroscopes. The gyroscopic bias may also not be known for a particular system ahead of time. The drift values and instability can be many different magnitudes depending on the price of the IMU, which is usually determined by the application. Some examples of these specifications according to application and price can be seen in Figure 6. As IMUs become more accurate and stable their prices increases. Luckily, this thesis focuses on a space based system which will usually have an expensive highly stable IMU [9].

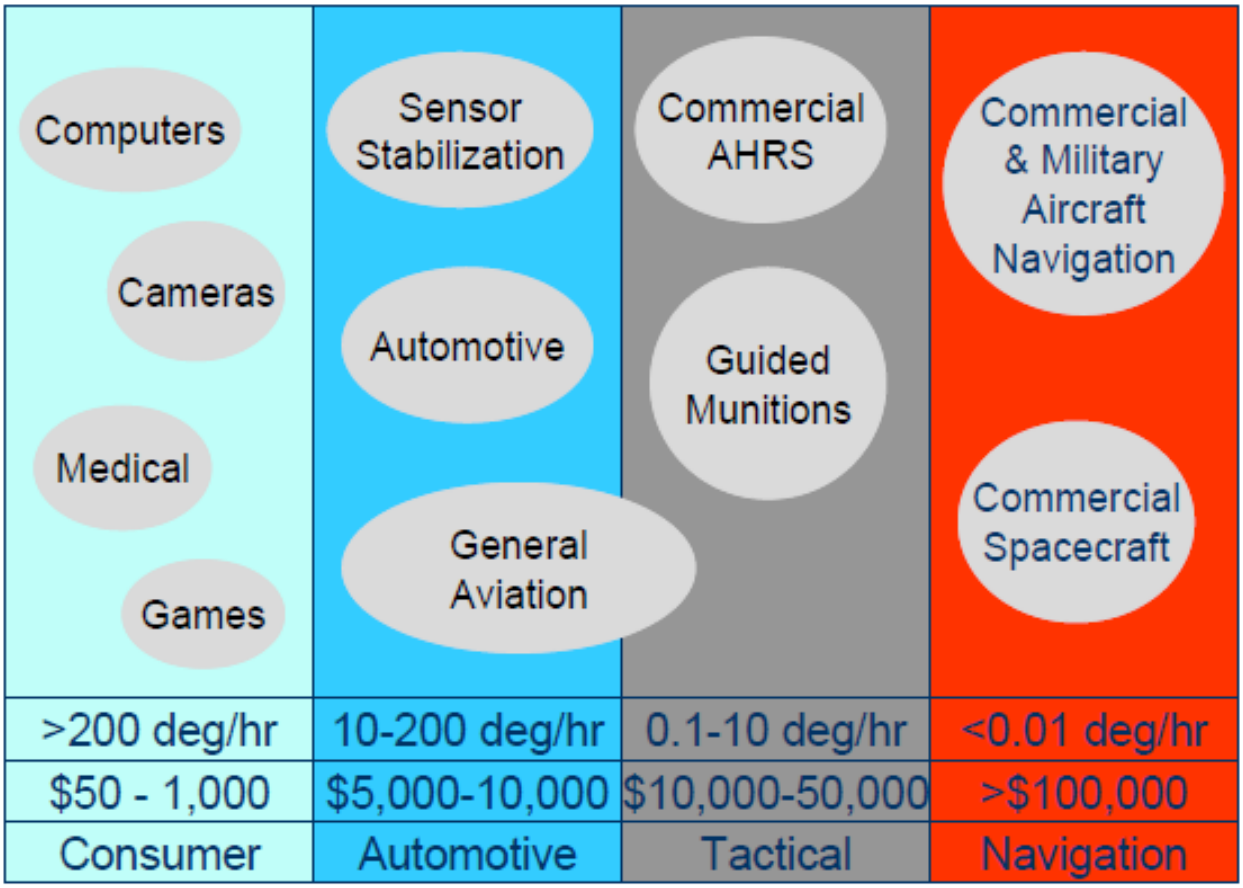

Figure 6. Gyroscope application specification magnitude [9].

Gyroscope to Sensor Error Transformation.

A linear error in the gyroscope will be much more useful if it transforms into a linear LOS measurement error. The geometric representation of this transformation can be seen in figure 7 where D_1 is the distance between the gyroscope and the sensor and D_2 is the distance between the sensor and the object. G represents the Gyroscope, ST represents the true sensor position, SF represents the false sensor position, incorrectly estimated by the gyroscope, and O represents the object which the sensor is tracking. The true measurement given the measured position of the sensor, according to the figure, would be the flat pointing vector in between SF and O. The false measurement, between the true sensor position and the object, contains the measurement error θ_{ME} induced by the

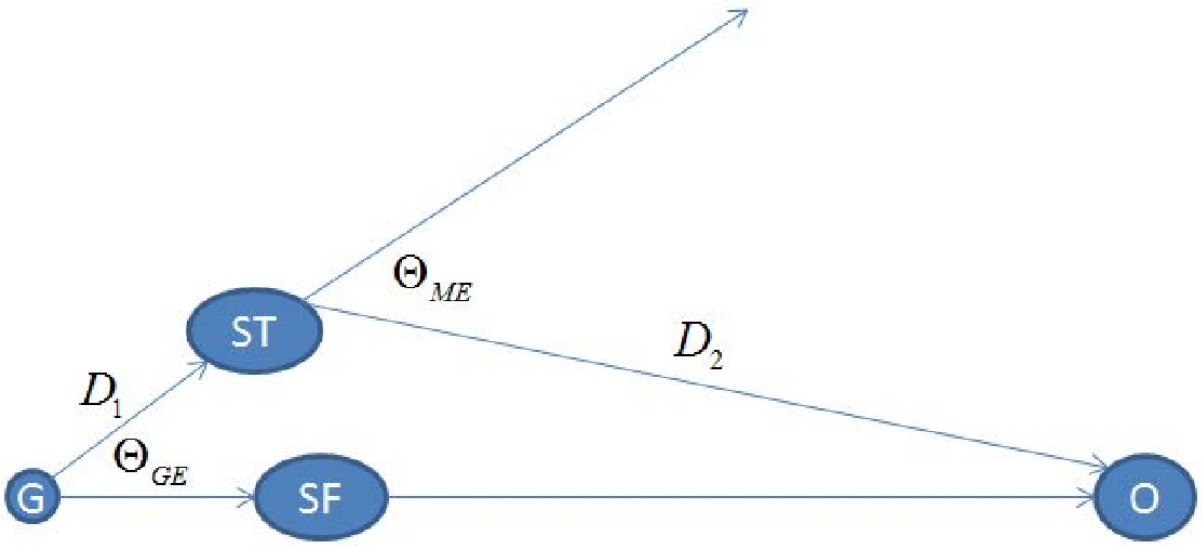

Figure 7. Gyroscopic error transformed to sensor error.

gyroscopic error θ_{GE}.

Solving for the relationship in between θ_{GE} and θ_{ME} produces

$$\theta_{ME} = \theta_{GE} + sin^{-1}\left(\frac{D_1 sin\theta_{GE}}{D_2}\right) \tag{3}$$

While this expression will not be purely linear, it will be a safe assumption that the gyroscopic errors will be small and the distance in between the gyroscope and the sensor will be much smaller than the distance between the sensor and the object. These assumptions allows a sin small angle approximations to be made, which turns (3) into $\theta_{ME} = \theta_{GE} + \frac{D_1\theta_{GE}}{D_2}$. Because of this, θ_{ME} will be a linear transform of θ_{GE}.

IMU Re-calibration.

Given a satellite application, the type of IMU drift described by Woodman could create an initially biased LOS measurement at the beginning of the sensing period. In [10],

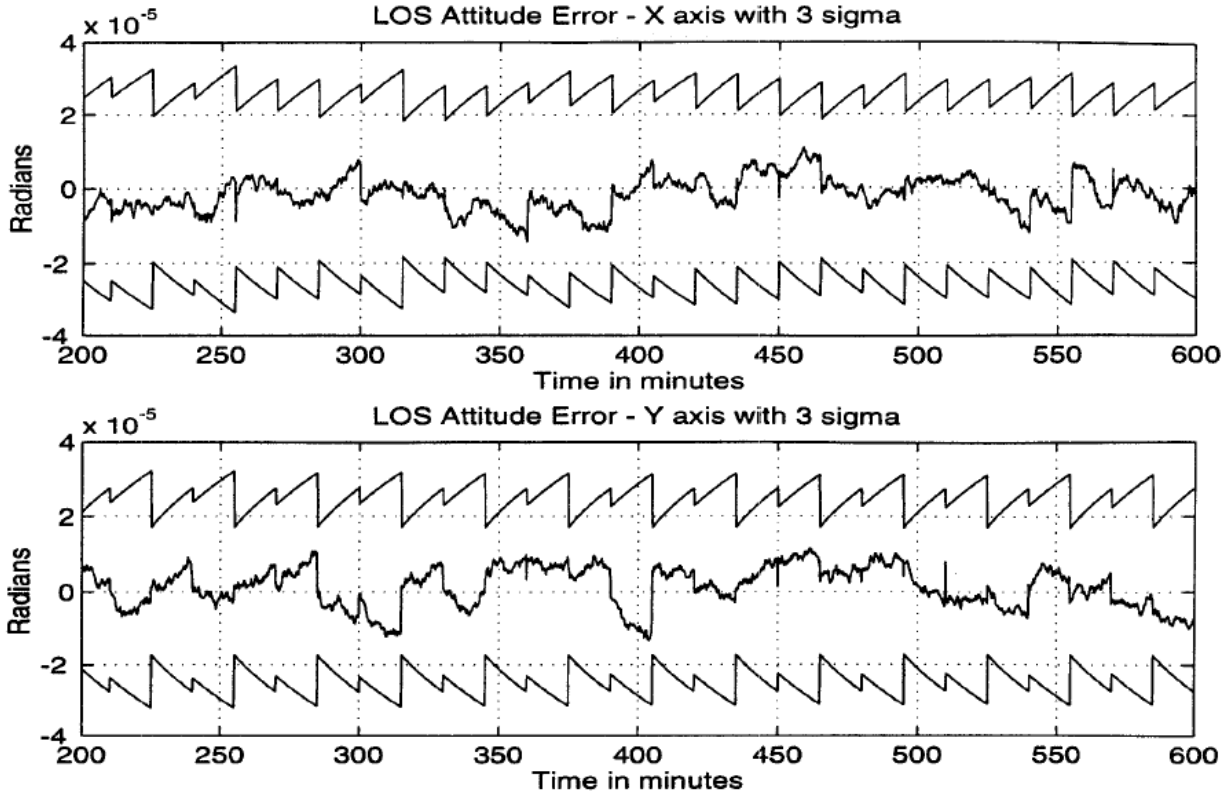

Figure 8. LOS Attitude Determination Errors - HEO Mission [10].

a satellite orientation star calibration method is developed. By optically observing the position of fixed stars in the sky, satellites can periodically re-calibrate to a more accurate global orientation measurement. In this paper the drifting error described by Woodman is simulated for a Highly Elliptical Orbit (HEO) orbit using this method. Every fifteen minutes the satellite is re-calibrated and the bias and drift error in the sensor is removed. An example of one of the results from this test can be seen in Figure 8. The LOS error was not allowed to drift far, but each star calibration was not necessarily perfect either [10].

The NLO algorithm, that has been developed by Sprang and Hartzell [1],[2], would work extremely well given a zero-mean Gaussian measurement error. Because of the LOS error described by Wu, a new method will be developed in this thesis for this biased and time drifting satellite attitude error. Assuming the LOS measurements will not overlap a re-calibration period, the attitude error can be described by an initial bias and a linear time

drift as well as an additive zero-mean Gaussian noise as described by Woodman. The new NLO will estimate the IMU bias and drift in conjunction with the trajectory for each sensor. It will be done similar to the way the receiver clock error is estimated in conjunction with the position in a GPS solution. This new method will be discussed in detail in the Methodology chapter of this thesis [7],[10].

2.3 Geolocalization Algorithms

Geolocalization algorithms tend to share many similar elements. These algorithms use data from multiple receivers or transmitters and calculate the position Maximum Likelihood Estimate (MLE) of an unknown transmitter or receiver from that data. The accuracy of the estimate will depend on the error within the system, the geometry of the system and the accuracy of the model used to create the estimate.

Geolocalization Bias Correction.

[11] develops an approach to bias estimation in a multiple sensor radar geolocalization. According to Portas, when tracking moving objects with biased sensors, the sensor fusion "...produces track instabilities ('zig-zag' effect due to erroneous maneuver detection in adaptive tracking filters), and may even produce multiple tracks (continuity faults) when systematic errors are higher than spatial correlation gates" [11].

GPS is a well-known geolocalization system and many elements are mathematically analogous to the NLO algorithm used in this thesis for AOA data. GPS is also a geolocalization system that corrects for a sensor bias, in this case, receiver clock error. GPS receivers receive pseudo-range estimates from multiple satellites. These pseudo-ranges are created by using the Time-Difference of Arrival and then calculating an

estimated range between the receiver and satellite. In order to create a position estimate and calculate the receiver clock error, at least four pseudo-ranges are needed. There are usually more than four pseudo-ranges to make the position estimate. Instead of ignoring the extra information, all of the pseudo-range estimates are used to find a more accurate estimate. All of the data is combined by using an iterative NLO algorithm to solve $\mathbf{A}x = B$ where B is a vector of all measurements, A defines a system for a particular model and x is the vector of estimates being made according to the system. The derivative of $\mathbf{A}x = B$ becomes $\Delta PR = \mathbf{J}\Delta\mathbf{X}$ where $\mathbf{J}$ is the Jacobian of the matrix of partial derivatives of the pseudo-range measurements with respect to the position estimate, $\Delta\mathbf{X}$ is the change in estimated position and clock error and ΔP is the change in pseudo-range. Given an arbitrary initial location guess, iterating

$$\begin{bmatrix} \Delta PR_1 \\ \Delta PR_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} ax_1 & ay_1 & az_1 & -1 \\ ax_2 & ay_2 & az_2 & -1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \Delta\widehat{x} \\ \Delta\widehat{y} \\ \Delta\widehat{z} \\ c\Delta t \end{bmatrix} \tag{4}$$

which will converge to the position MLE as $\widehat{X}_{new} = \widehat{X}_{old} + \Delta X$. This iterative process is called Newton's method [12]. This NLO method has been implemented for AOA measurements in Sprang and Hartzell; however, bias estimation has not been included [1],[2]. In this thesis a bias estimation in addition to the previous AOA Newton's method will be developed.

Geolocalization Confidence.

Another integral aspect of Geolocalization is defining position estimation confidence. According to Kay, the confidence of any type of estimate can be expressed by the CRLB. The CRLB can be found by inverting the Fisher information matrix shown in

$$I(\theta) = -E\left[\frac{\partial^2 ln(p(x;\theta))}{\partial\theta^2}\right] \tag{5}$$

where $p(x;\theta)$ is the PDF of the random process given parameters θ. The Fischer information matrix expresses the amount of information given the second derivative of the log likelihood function. The CRLB expresses the lower bound of variance on an estimate given the calculated information matrix [13]. According to Yang and Scheuing, the CRLB can be used in geolocalization problems to calculate the estimation variance of a geolocalization sensor configuration. Ideally, geolocalization sensors would create orthogonal LOS angles of intersection to create the smallest possible CRLB and, therefore, the smallest amount of error [14].

While the CRLB is the normal standard for creating a geolocalization confidence estimate, a CRLB with uncorrelated measurements is not the best method for the proposed geolocalization problem described in this thesis. While the measurements from different sensors will not be correlated, measurements from the same sensor will have some unknown correlation. The CI algorithm is a way to make estimates when the measurement correlation is unknown [4],[5]. Covariance intersections include these unknown correlations in the confidence ellipsoids. As seen in Figure 9, where the dashed blue lines represent the potential variance on the LOS measurements, the normal Covariance Ellipse produced by the CRLB could be much smaller and more over confident compared to the CI ellipse which takes into account potentially unknown measurement correlations. This

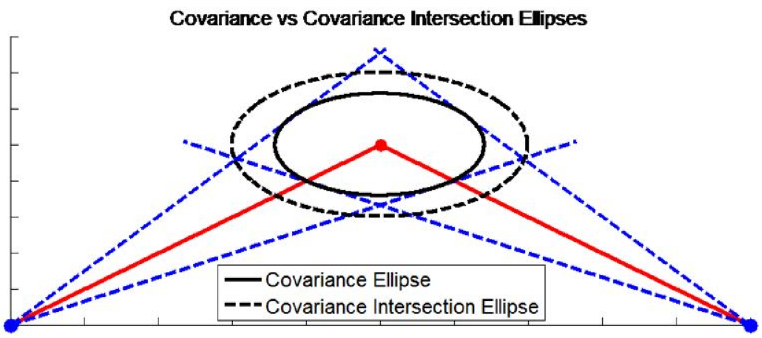

Figure 9. Covariance and Covariance Intersection Ellipses [1].

algorithm has been applied to the proposed problem in Sprang and will be used to assess confidence measurements in this thesis [1].

2.4 Confidence Credibility Metrics

Creating accurate confidence estimates is an integral part of object tracking and confidence credibility is a subject that cannot be ignored. A confidence credibility metric is a way of characterizing the statistical accuracy of a confidence model. In *Estimation with Application to Tracking and Navigation*, Shalom defines the metric NEES as defined in

$$NEES = (X - \hat{X})^T \mathbf{\Sigma}_X^{-1} (X - \hat{X}) \tag{6}$$

where the covariance matrix is inverted and multiplied by the estimate error. The NEES is something that can only be calculated in simulation when the true error is known. Averaging the NEES of each estimated point for a trial will produce the Averaged Normalized Estimation Error Squared (ANEES). Given a Gaussian covariance and the error vector, the NEES will be a chi squared distribution with three degrees of freedom

representing the three dimensional space in which the estimates are made. Validating three degrees of freedom will be simple given a chi-squared distribution. The degrees of freedom is equivalent to the expected value of the chi-squared distribution [15].

The NEES is a very useful metric; however, according to [16], it could have some problems. One of the largest potential problems that could affect the use of the NEES is that measuring the degrees of freedom from the NEES assumes that the estimation error is in fact Gaussian. If the error or covariance is not Gaussian then the NEES cannot be represented by a chi-squared distribution. There are potentially many non-Gaussian error elements that can push the error or covariance away from the central limit theorem to include sensor bias and drift, sensor covariance, the effect of sensor geometry on the positional error distribution, and estimating higher order terms of the object kinematic model. Many of these factors remain very close to a Gaussian like error and will hopefully remain close enough so that a Gaussian based approach to creating confidence ellipses is a safe approach. Li also mentions that using NEES as a metric favors pessimism much more than optimism. In this three-space application of the NEES, a very pessimistic confidence ellipse would be too large and would create a low NEES, while an optimistic confidence ellipse would be too small and create high NEES values. For pessimistic confidence estimates, a simulation with one estimate with large error will not affect the ANEES metric as much as it will for an optimistic confidence estimate. Because of this problem, it is tough to compare some NEES values [16]. How do you distinguish between pessimistic confidence estimates with one large error estimate and an optimistic confidence estimate without large error estimates? These large error estimate outliers are impossible to avoid, they are low probability, but something that can easily drive the ANEES far above its true value. A method to eliminate this problem will be discussed in chapter III.

2.5 Optimization Techniques

The NLO method has been applied to AOA measurements in Hartzell and Sprang. Instead of pseudo-ranges, AOA measurements are used. Newton's method is still used and converges to the position MLE. Hartzell and Sprang apply Newton's method more extensively to make velocity and acceleration estimates, as well as position estimates. The new $\Delta\Omega = J\Delta\mathbf{X}$ equation can be seen in

$$\begin{bmatrix} \delta\theta_1 \\ \delta\phi_1 \\ \delta\theta_2 \\ \delta\phi_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \frac{\delta\theta_1}{\delta x} & \frac{\delta\theta_1}{\delta y} & \frac{\delta\theta_1}{\delta z} & \frac{\delta\theta_1}{\delta \dot{x}} & \frac{\delta\theta_1}{\delta \dot{y}} & \frac{\delta\theta_1}{\delta \dot{z}} & \frac{\delta\theta_1}{\delta \ddot{x}} & \frac{\delta\theta_1}{\delta \ddot{y}} & \frac{\delta\theta_1}{\delta \ddot{z}} \\ \frac{\delta\phi_1}{\delta x} & \frac{\delta\phi_1}{\delta y} & \frac{\delta\phi_1}{\delta z} & \frac{\delta\phi_1}{\delta \dot{x}} & \frac{\delta\phi_1}{\delta \dot{y}} & \frac{\delta\phi_1}{\delta \dot{z}} & \frac{\delta\phi_1}{\delta \ddot{x}} & \frac{\delta\phi_1}{\delta \ddot{y}} & \frac{\delta\phi_1}{\delta \ddot{z}} \\ \frac{\delta\theta_2}{\delta x} & \frac{\delta\theta_2}{\delta y} & \frac{\delta\theta_2}{\delta z} & \frac{\delta\theta_2}{\delta \dot{x}} & \frac{\delta\theta_2}{\delta \dot{y}} & \frac{\delta\theta_2}{\delta \dot{z}} & \frac{\delta\theta_2}{\delta \ddot{x}} & \frac{\delta\theta_2}{\delta \ddot{y}} & \frac{\delta\theta_2}{\delta \ddot{z}} \\ \frac{\delta\phi_2}{\delta x} & \frac{\delta\phi_2}{\delta y} & \frac{\delta\phi_2}{\delta z} & \frac{\delta\phi_2}{\delta \dot{x}} & \frac{\delta\phi_2}{\delta \dot{y}} & \frac{\delta\phi_2}{\delta \dot{z}} & \frac{\delta\phi_2}{\delta \ddot{x}} & \frac{\delta\phi_2}{\delta \ddot{y}} & \frac{\delta\phi_2}{\delta \ddot{z}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \delta\widehat{x} \\ \delta\widehat{y} \\ \delta\widehat{z} \\ \delta\widehat{\dot{x}} \\ \delta\widehat{\dot{y}} \\ \delta\widehat{\dot{z}} \\ \delta\widehat{\ddot{x}} \\ \delta\widehat{\ddot{y}} \\ \delta\widehat{\ddot{z}} \end{bmatrix} \tag{7}$$

where Ω is the LOS measurement matrix, $\mathbf{X}$ is the complete state estimation vector, $\dot{x}, \dot{y}, \dot{z}$ are the ECEF cartesian velocity components and, $\ddot{x}, \ddot{y}, \ddot{z}$ are the ECEF cartesian acceleration components [2],[1].

The new NLO includes a weighted covariance matrix $\mathbf{\Sigma}_\Omega$ calculated by the equation $\mathbf{\Sigma}_\Omega = (\mathbf{W}^T\mathbf{W})$, where $\mathbf{W}$ is the covariance intersection weighting matrix made up of weight parameters w. The derivation for each iterative step can be seen in

$$
\begin{aligned}
\mathbf{W}J(\mathbf{X}_{init})\Delta\mathbf{X} =& \mathbf{W}\Delta\Omega, \\
J^T\mathbf{W}^T\mathbf{W}J\Delta\mathbf{X} =& J^T\mathbf{W}^T\mathbf{W}\Delta\Omega, \\
\Delta\mathbf{X} =& (J^T\mathbf{W}^T\mathbf{W}J)^{-1}J^T\mathbf{W}^T\mathbf{W}\Delta\Omega, \\
\Delta\mathbf{X} =& (J^T\Sigma_\Omega^{-1}J)^{-1}J^T\Sigma_\Omega^{-1}\Delta\Omega. \qquad (8)
\end{aligned}
$$

Given an initial position, $\Delta\mathbf{X}$ will be recalculated until it diminishes below a set threshold.

Applying this method to a fast moving object complicates many different parameters of the algorithm. For instance, the amount of measurements used to make a single estimate can affect the algorithm's performance. The measurements used for each estimate is called the window size. Estimation accuracy is susceptible to the size of the windows. If the window is too small, the noise in the measurements can give an inaccurate estimate of the moving object. If the window is too large, the Jacobian grows extremely large and becomes computationally taxing. Each window could become ineffective if the measurements are primarily from one sensor. The timing of the measurements need to be adjusted as well. Each sensor will not observe the object at the exact same time or even at the same rate. A time transition matrix is used to adjust each of the measurements to account for this complication. The time delay in between the object and each sensor has to be accounted for as well. Given that this algorithm will be tested on satellite based sensors, the time delay will not always be negligible.

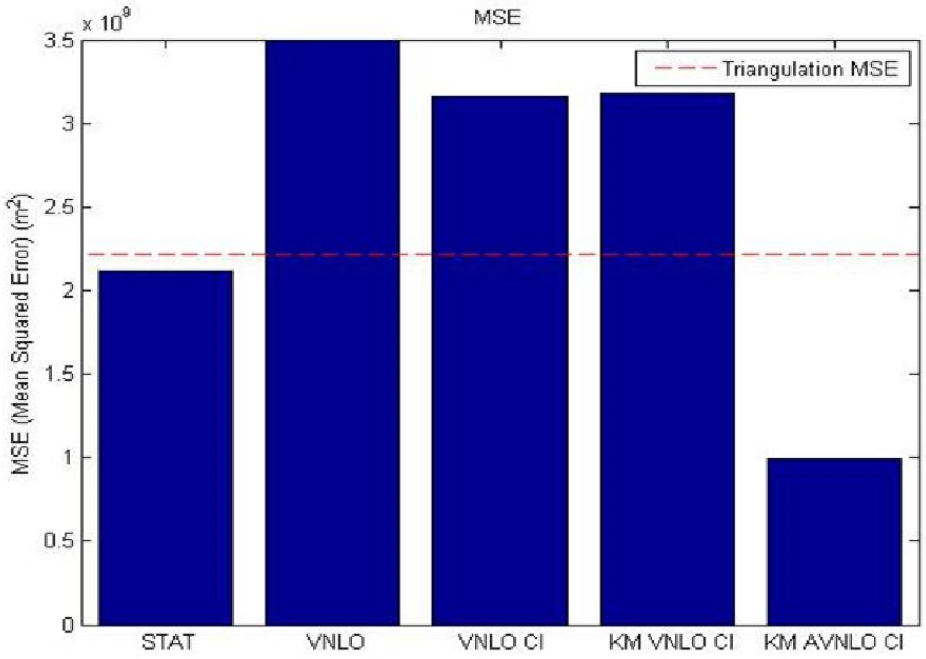

Figure 10. MSE from different AOA NLO algorithms tested by Sprang [1].

2.6 Past NLO Results

In Sprang, the above discussed NLO was tested with varying configurations of equation 7. A STAT or static NLO would only estimate the position and ignore any type of kinematic movement. A Velocity NLO (VNLO) optimizes over a constant velocity model for each point, a Velocity NLO with CI (VNLOCI) includes the CI in the NLO algorithm, a Kinematic Model VNLOCI (KMVNLOCI) optimizes VNLO estimates over an entire kinematic path model where constant velocity is assumed in between estimated points, and a KMAVNLOCI optimizes over a kinematic path model where constant acceleration is assumed in between estimated points. These NLO variations were tested against the triangulation algorithm in a Monte-Carlo Simulation. Figure 10 shows the resulting MSE from each of the different algorithms for a complicated (staged) path [1].

Figure 11 shows the resulting NEES values for each of the algorithms. The

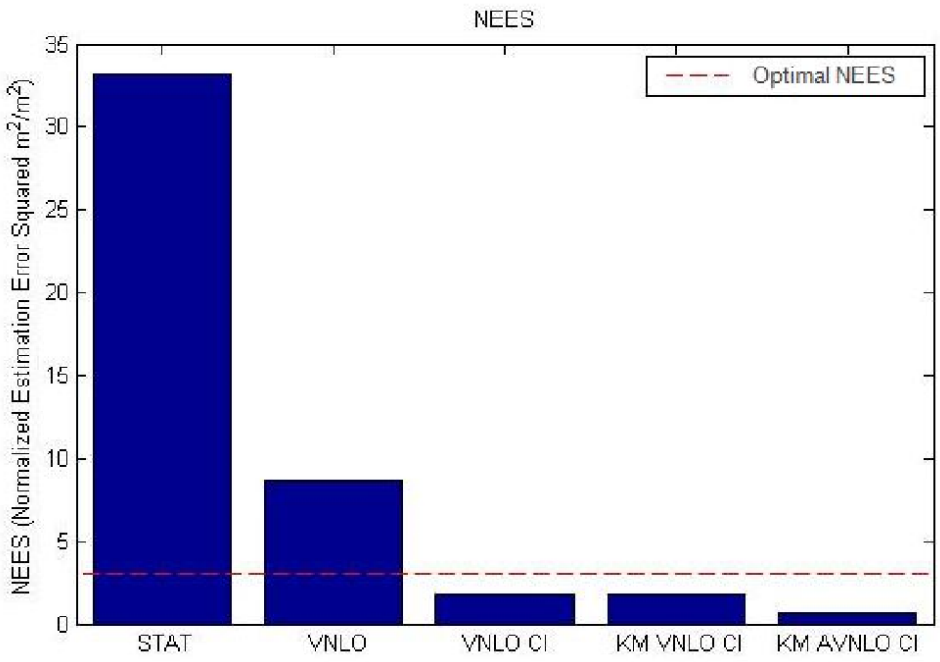

Figure 11. NEES from different AOA NLO algorithms tested by Sprang [1].

Monte-Carlo simulation shown in Figures 10 and 11 consisted of 25,000 iterations of complicated paths with varying simulated sensor error and geometric configurations. In terms of accuracy, the static NLO algorithm performed slightly better than triangulation, and adding more state elements did not improve complicated path estimates, except in the KMAVNLOCI where acceleration elements were estimated. NEES is a normalized metric to determine the confidence accuracy of an estimate. For a geolocalization algorithm, an ideal location NEES will be three because the estimate will be in 3-space. As shown in Figure 11, the static NLO and VNLO were over confident because they did not account for the unknown correlation between same-sensor measurement. Once CI was introduced to the VNLOCI, KMVNLOCI, and KMAVNLOCI algorithms, the unknown correlations were taken into account and the confidence estimates were under confident. Under confidence suggests that the pre-calculated error ellipsoids were larger than the produced ellipsoids from the Monte-Carlo Simulations [1]. In this thesis, the new modifications will be made and compared to the KMAVNLOCI algorithm.

2.7 Trajectory Modeling and Estimation

In many cases, the object trajectory being observed may change from one type of trajectory to another. In other words, at discrete moments, an object could quickly change its acceleration vector if there were a change in force or changing force vector on the object. If an estimation window were to overlap two different trajectory models, an estimate would be split between two different acceleration vectors and produce an inaccurate path model. The previously developed windowing method does not work well for discrete changes; however, if the path is segmented into easily digestible segments, the NLO algorithm can be optimally used on each segmentation separately. Panagiotakis and Lee discuss a trajectory clustering method to analyze and group trajectory data together. In their trajectory clustering method, there is a trajectory segmentation method which can be used to segment a single path into multiple paths categorized by path type. Each trajectory path can be defined by an initial position, initial velocity and acceleration. Of course, there will be error involved in the observed trajectory. The matrix representation of a trajectory model can be seen in

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \frac{1}{2}t_1^2 & t_1 & 1 \\ \frac{1}{2}t_2^2 & t_2 & 1 \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} a_0 \\ v_0 \\ x_0 \end{bmatrix} + \begin{bmatrix} error_1 \\ error_2 \\ \vdots \end{bmatrix} \tag{9}$$

$$X = H\theta + \varepsilon$$

where t is the time the position estimate $\begin{bmatrix} a_0 & v_0 & x_0 \end{bmatrix}^T$, X is the vector of position estimates, X_i is the estimated position, ε is the error vector and $\mathbf{H}$ is the time transition matrix. While 9 shows the trajectory definition in 1D, it can be repeated separately for each dimension estimated [17][18].

In order to bypass this problem, a MLE will be used to develop this algorithm. In this case, an NLO will not be necessary to solve. Instead, it can be solved in closed form. According to Kay the MLE of the linear estimator 9 in the form $X = \mathbf{H}\theta + \varepsilon$ can be expressed by

$$\widehat{\theta} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T X \tag{10}$$

where $\widehat{\theta}$ is the MLE of trajectory vector. Because the trajectories used in this thesis will be noisier than those discussed by Panagiotakis and Lee, the algorithm will be modified to make over determined estimates. The trajectory segmentation algorithm will be developed in the next chapter of this thesis in detail [13].

2.8 Conclusion

All of the theoretical background in this chapter is necessary for the understanding and development of the AOA NLO algorithm. The NLO algorithm can use inspiration from many different algorithms, especially when characterizing a path as complex as an object with a discretely changing trajectory model. The quality of sensor characterization will determine how effective the algorithm performs in an operational setting. If an algorithm is designed for the wrong error distribution, an unknown bias or large and uneven variance can form in the MLEs. This thesis is the third iteration of a NLO satellite based AOA algorithm project. For the last iteration of this project, many of the previous assumptions have been fully justified or fixed for a fully robust AOA NLO algorithm. This thesis will also validate and further analyze tests that have been done on the AOA NLO by Sprang and Hartzell.

III. Methodology

3.1 Introduction

The baseline algorithm developed in this chapter will be the NLO algorithm described and developed in detail by Sprang and Hartzell. While Hartzell focused on developing a basic NLO algorithm and Sprang focused on creating a more accurate confidence estimate, as well as developing NLO algorithms including new object kinematic models, the new modification will focus on increasing accuracy by segmenting complicated paths and accounting for biased and time drifting sensor error. The algorithm developed by Sprang and Hartzell will be treated like a module in the new algorithm [1] [2]. Accuracy will also be increased by estimating the bias and drift of the sensors estimated from the AOA measurements. The Bias and Drift estimate will be built on the assumption of some linear error function in the sensors. As well, a method was developed and tested to segment the trajectory of a complicated path into easily processed sections.

3.2 Gaussian Model Validation Tests

AOA estimates are not actually Gaussian but under certain conditions they can be assumed Gaussian. There are two different variables which can make an AOA localization estimate Gaussian or non-Gaussian measurement angle error variance and the angle of intersection between LOS vectors or parallax measurements. Not accounting for the effect of the localization algorithm on the resulting distribution, these two variables can be tested to find the range in which AOA localizations are Gaussian and when it is theoretically impossible. In order to clearly define the limit, a two dimensional simulation between two sensors will sweep through all possible angles of intersection and estimate the maximum

amount of variance sensors can have before the localizations stop being Gaussian. At each angle, simulated variance will be decreased incrementally until the data is observed to be Gaussian using Trujillo-Ortiz's Royston test function. The Royston test is a normal H0/H1 p-value test that incorporates a Shapiro-Wilk test for a multi-variate data set. As long as the data set is small enough and each data point is uncorrelated, the Royston test performs well with a excellent Type I error control [6].

To find the test statistic determining if a data set is Gaussian, the Royston's test first finds the Shapiro-Wilk statistic W_j for each variable in the tested p-variate distribution. After calculating g, m, and s from polynomial approximations based on the number of data points, R_j is calculated as seen in

$$R_j = \left\{ \phi^{-1} \left[\frac{1}{2} \phi \left\{ -((1 - W_j)^g - m)/s \right\} \right] \right\}^2, \tag{11}$$

where ϕ is a standard Gaussian CDF. H, shown as

$$H = e \sum_{j=1}^{p} \frac{R_j}{p}, \tag{12}$$

is Chi-square distributed if the original data is multi-variate Gaussian. If H is in fact Chi-square distributed, it will have e degrees of freedom expressed by

$$e = \frac{p}{1 + (p - 1)mC}, \tag{13}$$

where mC is the average correlation estimated from the R_j's. Once the data is transformed into a chi-square distribution with an estimated degrees of freedom, the chi-square goodness of fit test probability using the MatLab *chi2cdf* function.

$1 - chi2cdf(H, e)$ produces the p-value that the original data set is in fact multi-variate Gaussian. These results will be covered in chapter IV [6].

3.3 Trajectory Segmentation

In this section, a robust Trajectory Segmentation algorithm will be developed. Methods shown in [17] were considered and altered to create a split and merge type segmentation algorithm. A flow chart describing the method developed can be seen in Figure 12. The algorithm contains a split and a merge section. In the split section, the algorithm checks the trajectory data against a pre-set residual. If the cost of assuming a one-model trajectory is larger than a set threshold, the algorithm will look for the most likely location of a split. Next, the split algorithm will be applied to each new trajectory segment. This loop will continue until all splits are found and all line segments are below the pre-set threshold. The split section alone will successfully perform the segmentation required for the NLO optimization; however, there is a large false alarm rate unless very precise thresholds are chosen. A merge section is added to reduce the false alarm rate. In the merge section of the algorithm, each split is evaluated by a line estimator and the two segments of the split will merge if the cost of no split is less or significantly similar to the cost of a split. This split and merge method can be applied to any element of the trajectory whether it is the position, velocity or acceleration or any combination. While there will be less noise on the position or velocity, these elements may not contain easily detected discrete changes as the higher order acceleration or jerk elements. This trajectory segmentation method will be developed so that it can be applied to any trajectory element.

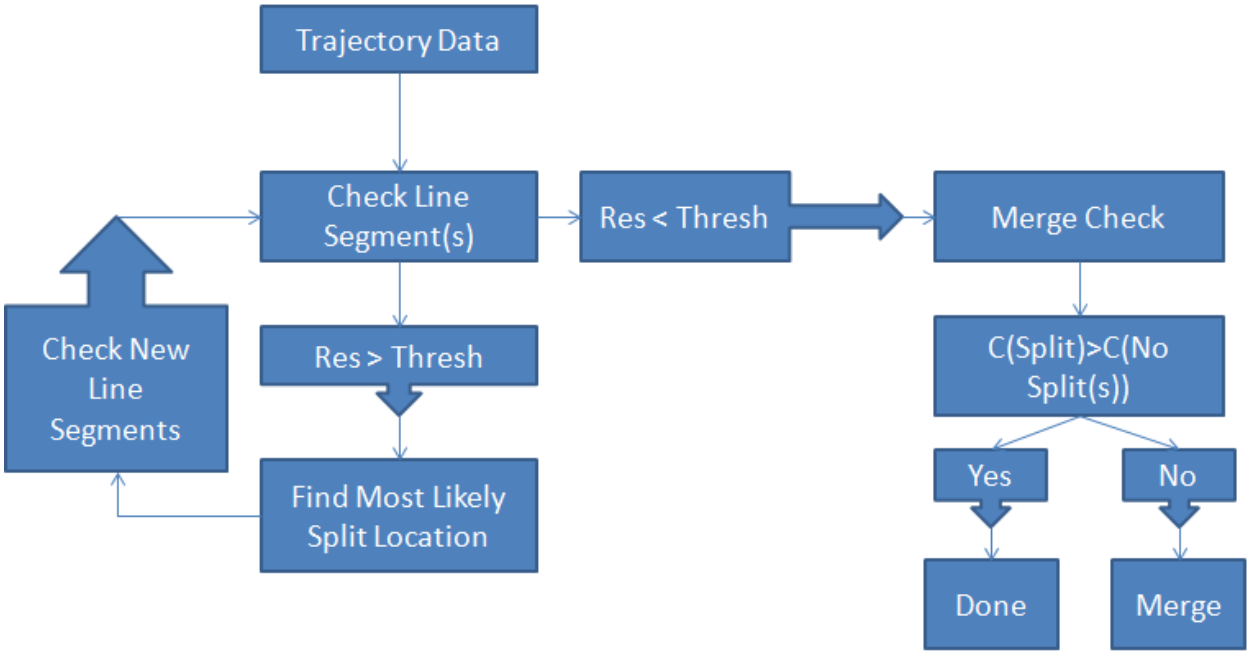

Figure 12. Trajectory Segmentation Flow Chart.

Line Segment Check.

Given a vector of positions and corresponding measurement times, nth-order trajectory elements can be estimated. The time and position vectors are transformed into the linear system shown in (14). According to [13], a linear system with noise can be estimated in the following equations shown in

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} \frac{1}{2}t_1^2 & t_1 & 1 \\ \frac{1}{2}t_2^2 & t_2 & 1 \\ \frac{1}{2}t_3^2 & t_3 & 1 \\ \frac{1}{2}t_4^2 & t_4 & 1 \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} a_0 \\ v_0 \\ x_0 \end{bmatrix} + \begin{bmatrix} error_1 \\ error_2 \\ error_3 \\ error_4 \\ \vdots \end{bmatrix}$$

$$\widehat{\theta} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{X}. \tag{14}$$

Next, the estimate residual can be calculated by taking the 2-norm of the difference between the trajectory from the given time matrix H and the estimated trajectory elements with the observed trajectory as seen in

$$residual = \left|\left|\mathbf{H}\widehat{\theta} - X\right|\right| \tag{15}$$

This process can be repeated across n dimensions to create a cost metric for a n_{th} order one model trajectory estimate. Even with inherent noise in the observed trajectory, a one trajectory model estimate on a trajectory with only one model will be much lower than a one-model estimate of a multiple model trajectory [13].

Split Location Estimate.

Once it is observed that a trajectory segment does not fit a specific trajectory model, the split location is found. The split location module first makes a windowed trajectory model estimate across the entire segment. For each window, the estimate cost residual is

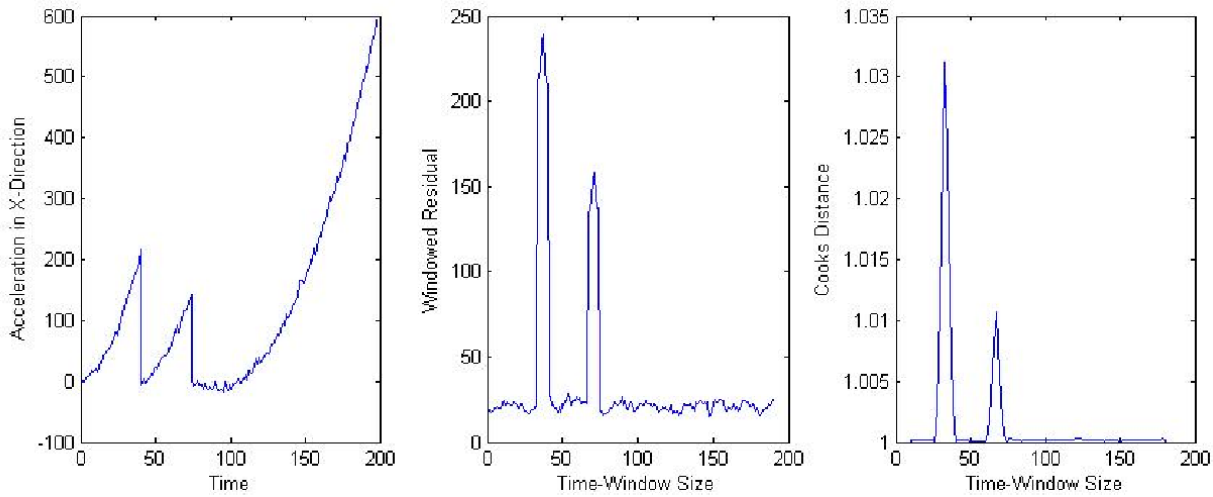

Figure 13. 1D split localization example.

calculated and each window iteration forms a new residual vector. Next, a windowed Cook's distance is taken of the residual matrix to normalize the non-split locations to a number close to one and further highlight the locations of the observed splits. Cook's distance is in essence an outlier detector, it is calculated by finding the change in mean of data when a specific data point or data points are taken out of the data. The maximum Cook's distance will occur at the most likely location of the split. As seen in Figure 13 the residual calculation spikes at the split locations and is further smoothed by the Cook's distance for a more exact split localization.

Merge Check.

The split detection example shown in Figure 13 contains a very clear split. When the splits are clear and obvious the split threshold can be higher which prevents many false alarms. In many useful scenarios, the thresholds used need to be much lower to detect all of the splits when noise of the observed path increases and the model change is smaller or less discrete. When thresholds are lowered to detect all of the paths, many false splits are detected. The easiest solution to this problem is to compare the cost of a two model

trajectory with a one model trajectory for each split. The two types of models can be done with the same residual check shown in (14) and (15) where the one model residual is multiplied by a cost factor associated with complicating a trajectory model with more segments.

3.4 Bias and Drift Estimation

In this section a bias and drift estimator algorithm will be developed based on Sprang and Hartzell's AOA NLO algorithm. There are two different ways a bias and drift estimator can be implemented with the AOA NLO algorithm. It could be included in the Jacobian matrix within each iterative step of the NLO. The advantage of including the bias and drift estimate in the Jacobian is that the algorithm will run only once and could potentially converge to an exact solution. Because this modification is a new development of the AOA NLO algorithm, it will be referred to as the BDNLO algorithm. The other bias and drift estimator developed in this thesis will be an iterative approach where the NLO algorithm is treated like a module, where the bias and time drift are estimated in post processing. Once the bias and drift are estimated, they are applied to the observed LOS data and the NLO applied again to the corrected LOS data. This cycle iterates until the new bias and drift estimates are close to zero and do not significantly change the LOS measurements. This method will treat the KMAVNLOCI as a module and will be referred to as the BDMOD algorithm.

NLO based Bias and Drift Detection.

In the NLO based approach or BDNLO, the Jacobian is heavily modified to include $4 \times NS$ columns where NS represents the number of sensors used in the estimation. The

columns are added to the right of the normal Jacobian and the estimation vector as seen in

$$\mathbf{X} = \begin{bmatrix} x & \dots & \ddot{z} & B\theta_1 & D\theta_1 & B\phi_1 & D\phi_1 & B\theta_2 & \dots \end{bmatrix}^T \tag{16}$$

and

$$\begin{bmatrix} \delta\theta_1^1 \\ \delta\phi_1^1 \\ \delta\theta_2^2 \\ \delta\phi_2^2 \\ \delta\theta_3^1 \\ \delta\phi_3^1 \\ \vdots \end{bmatrix} = \begin{bmatrix} \dots & 1 & t_1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 1 & t_1 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 1 & t_2 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & 0 & 1 & t_2 & \dots \\ \dots & 1 & t_3 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 1 & t_3 & 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \delta B\theta_1 \\ \delta D\theta_1 \\ \delta B\phi_1 \\ \delta D\phi_1 \\ \delta B\theta_2 \\ \delta D\theta_2 \\ \delta B\phi_2 \\ \delta D\phi_2 \\ \vdots \end{bmatrix} \tag{17}$$

where B is the bias element and D is the linear time drift element for a particular sensor elevation or azimuth measurement. With the new Jacobian and estimator matrices, the NLO is solved the same as described in Sprang and Hartzell [1],[2]. This estimation method is very similar to a GNSS solution as seen in Kaplan [12]. However, the Jacobian in this AOA NLO method is much larger than a GNSS NLO Jacobean. The Jacobian is also a very different form with many zeros on the right side of the matrix. The form of this matrix creates a potential problem caused by the computational resources for this project. Many large matrices with large and small values have large condition numbers. The condition number is a metric that expresses how difficult it is to calculate the inverse of a

particular matrix given the numerical precision of the computer as described by Trefethen [19]. In each step of the NLO the inverse of $J^T J$ is taken. The old NLO $J^T J$ matrix is a $9x9$ matrix for a full acceleration, velocity and position estimate. In the proposed new algorithm, the $J^T J$ will be a $9 + 4 \times \text{NS}x9 + 4 * \text{NS}$ sized matrix that will not be heavily diagonal. The proposed matrix has shown to create large condition numbers for very simple NLO estimates. Containing a large condition number not only makes an inversion computationally limited, but it also suggests the NLO is less likely to converge. The NLO based bias and drift estimation would be more effective on a system using a more precise computational environment with 128 bit numbers instead of 64 bit. This algorithm will be tested; however, the convergence rate is not expected to be very high compared to the other AOA NLO algorithms due to the computational limitations.

Post Processing Bias and Drift Detection.

The second way to estimate the bias and drift is through a linear estimator after the NLO algorithm has converged to a stable solution. The BDMOD uses the KMAVNLOCI as a module and estimates the sensor bias and time drift afterwards. By taking the difference between the observed LOS measurements and the LOS that should be observed given the position solution of the algorithm $\Delta\Omega$, a linear $\Delta\Omega = \mathbf{H}X + \varepsilon$ system can be written as seen in

$$\begin{bmatrix} \theta_1^2 \\ \phi_1^2 \\ \theta_2^1 \\ \phi_2^1 \\ \theta_3^1 \\ \phi_3^1 \\ \vdots \end{bmatrix} - \begin{bmatrix} \widehat{\theta_1^2} \\ \widehat{\phi_1^2} \\ \widehat{\theta_2^1} \\ \widehat{\phi_2^1} \\ \widehat{\theta_3^1} \\ \widehat{\phi_3^1} \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & t_1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & t_1 & \dots \\ 1 & t_2 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & t_2 & 0 & 0 & 0 & 0 & \dots \\ 1 & t_3 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & t_3 & 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} B\theta_1 \\ D\theta_1 \\ B\phi_1 \\ D\phi_1 \\ B\theta_2 \\ D\theta_2 \\ B\phi_2 \\ D\phi_2 \\ \vdots \end{bmatrix} + \begin{bmatrix} error_{\theta 1} \\ error_{\phi 1} \\ error_{\theta 2} \\ error_{\phi 2} \\ error_{\theta 3} \\ error_{\phi 3} \\ \vdots \end{bmatrix}$$

$$\Delta\Omega = H\mathbf{X} + \varepsilon. \tag{18}$$

where $\mathbf{H}$ is a sensor measurement populated first order time transition matrix, and θ_i^j is the i_{th} measurement which was observed from sensor j. By making a $(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\Delta\Omega$ MLE of $\mathbf{X}$ as described by Kay, the bias and drift are able to be estimated [13]. While this estimate will still use a very large matrix inversion, it is not as uneven as the required NLO matrix and it is a non-iterative estimate so the matrix condition number is no longer as large of a requirement for convergence. After the bias and drift are estimated, the LOS measurements are adjusted and a new localization solution calculated from running the NLO again with the new measurements. This process is repeated until there is no more bias and drift detected in the system.

The disadvantage of this method is that it runs more slow by applying the KMAVNLOCI multiple times. The other potential problem, is that the NLO algorithm needs to produce a position estimate close enough to the true trajectory for a useful bias

and drift estimate. Error in the initial trajectory estimation could produce an even more inaccurate bias and drift estimation. Due to this error propagation, this algorithm likely produces a less accurate estimate than the BDNLO described in the previous section. The speed of the algorithm is not a large concern in this thesis so the slower performance is not a problem. The problem regarding the required accuracy of the KMAVNLOCI is be tested by Monte Carlo simulations and presented in Chapter IV. The specifics of these tests is described later in this chapter.

3.5 Covariance Intersection

In order to account for statistical correlation in between measurements from the same sensor in the state estimate and the confidence estimate, CI is used. This method has been developed and tested by Sprang. In order to include the CI algorithm in the Jacobian method, we need

$$\Delta X = (J^T(\mathbf{W}\mathbf{\Sigma}_\Omega^{-1})J)^{-1}(J(\mathbf{W}\mathbf{\Sigma}_\Omega^{-1})\Delta\Omega). \tag{19}$$

is used as each NLO step, where $\mathbf{\Sigma}_\Omega$ is the sensor variance matrix and $\mathbf{W}$ are a pre-determined weighting matrix. $\mathbf{\Sigma}_X$, expressed in

$$\mathbf{\Sigma}_X^{-1} = J^T(\mathbf{W}\mathbf{\Sigma}_\Omega^{-1})J. \tag{20}$$

will be the covariance matrix. The Cholesky composition of Σ_X^{-1} will produce a location confidence estimate for each position estimate. More mathematical detail about the application and justification for this use of CI can be found in Sprang [1].

3.6 Simulation

In this section, the types of simulations are developed in detail. There are two important aspects of the simulation, the positions of the sensors, the object being tracked, and the error function in the sensor LOS measurements.

Object and Sensor Trajectory.

The object and sensor position simulations used in this thesis are the same that were developed by Sprang and Hartzell. The sensors being used are simulated LEO satellites. Each sensor will have a direct LOS to the object being tracked unobstructed by the Earth. The sensed object will have two different types of paths, simple and complicated. A simple path will be the same type of simulation as the sensors. Simple paths will be created by randomly generating orbital parameters for a non-elliptical LEO satellite to include, elevation, Longitude of ascending node, inclination, and true anomaly. Sprang and Hartzell tested a system where each sensor would be creating measurements evenly over the entire trajectory. In this thesis, that method will be tested as well as a random visibility test where at increments, some sensors will not produce measurements in set time periods. This will simulate each sensor finding and losing track of whatever object is being detected. A simple path simulation will create an object with a very steady and predictable pattern. The NLO algorithms are more likely to create much more accurate solutions when tracking the simple path which will be accurately estimated by the constant acceleration kinematic model in the KMAVNLOCI [1],[2].

A complicated path simulates an object with a constantly changing force vector. A constantly changing force on an object will create the worst case scenario for the NLO algorithm with a constant acceleration kinematic model. If the acceleration is changing

and the path of the object is extremely unpredictable, the constant acceleration model of the NLO is likely to have plenty of trouble and could produce inaccurate estimates. The NLO may not even converge if the path is complicated enough. The complicated path will be modeled the same was as it was in Sprang with the MATLAB Ordinary Differential Equations (ODE) solvers. The ODE solution will limit the object to movement that is physically possible for an orbital object. This movement does not necessarily match a specific type of object but will be unpredictable enough to test the upward limits of the NLO algorithms.

Gyroscopic Error Simulation.

The accuracy of the LOS measurements will be a direct cause of the simulated gyroscopic accuracy of each sensor. In this thesis, the gyroscopes will be simulated similar to the gyroscopes described Woodman, Wu, Gebre and Gautier [10],[7],[8],[9]. For each sensor θ and ϕ, there are three simulated elements, an initial bias, a linear drift and a drift stability. The initial bias, expressed by B, is the sensor error at the beginning of the simulation. The bias value is the result of the unknown integrated gyroscopic error. The bias is the largest source of error because it incorporates the total prior gyroscopic error that is unobserved. The drift, expressed by D, is the linear time drift of the of the sensor in radians error per second. The time drift will be assumed constant during the sensing period. The drift instability, expressed as DI, is the noise added to the simulated random walk. This instability could create second order walk functions and is expressed in $radians/second^{1/2}$. Each measurement error is always dependent on the measurement error simulated previously in time from its own sensor. The next or ii_{th} error is simulated by using

$$
\begin{aligned}
TD =& MeasurementTime(ii+1) - MeasurementTime(ii), \\
Error(0) =& B, \\
Error(ii+1) =& Error(ii) + \mathcal{N}(D * TD, (DI * TD^{1/2})^2).
\end{aligned} \tag{21}
$$

where $N(\mu, \sigma^2)$ is a normally distributed random number with mean μ, TD is the Time Difference between the measurements and variance σ [7]. The linear time drifting error estimator is not specifically made for zero mean random walk over a long time; however, the error estimator should work well for a random walk sampled across a small enough time period where the error appears to be nearly linear. A linear approximation will be a better estimator than a zero mean, zero drift approximation as long as the drift instability is low enough within the sensing time period. The gyroscopic error random walk is usually linear like for the amount of time the sensors will be simulated. The NLO algorithms is tested to see how linear the error needs to be before the NLO algorithms become ineffective along side the other error parameters, this test is presented and discussed in Chapter IV. The values for the three error parameters is uniformly distributed across a range close to the type of error simulated by Wu [10].

3.7 Algorithm Comparison

Evaluating the NLO algorithms requires accurate metrics describing the performance in an easy to understand way. Accuracy and confidence metrics are useful for assessing geolocalization performance. In this section, those metrics are discussed. For adequate testing of these two new additions to the NLO algorithm, two different types of tests are conducted. For the first tests, Monte Carlo trials are simulated where all variables

and paths tested will be different. A Monte Carlo assessment will build statistical metrics to assess how the algorithm performs on average. By the end of the Monte Carlo Trials, PDFs will be estimated to show the probability of positional accuracy as well as confidence accuracy against many different variables. The Monte Carlo simulations will be split in between simple and complicated paths due to the discretely different performance. Next a set of generic paths (simple or complicated) are created and every variable will be held constant except one. By controlling all but one element, the effect of each variable on all algorithms can be assessed separately.

Metrics Used.

The purpose of these algorithm alterations is to improve the accuracy of the previous NLO algorithms. The metrics should express the absolute accuracy error as well as a metric to determine if the calculated confidence statistically reflects the result. The absolute accuracy will be expressed in Two Norm Error (TNE), the confidence assessment will be expressed in NEES and to determine if the modifications are improving the NLO on a case by case level, the Increase in Percentage Accuracy (IPA) is also calculated. The three metrics can be shown as

$$TNE = \sqrt{(X - \hat{X})^T (X - \hat{X})}, \tag{22}$$

$$NEES = (X - \hat{X})^T \mathbf{\Sigma}_X^{-1} (X - \hat{X}), \tag{23}$$

$$IPA = \frac{NLO_{TNE} - ModifiedNLO_{TNE}}{NLO_{TNE}} \times 100. \tag{24}$$

TNE is calculated by taking the square root of the square of the difference between the true position state vector X and estimated state vector $\hat{X}$ as seen in (22). To calculated

the NEES, the covariance matrix $\mathbf{\Sigma}_X^{-1}$, expressed in (20) is used as seen in (23) to assess how well the covariance of the estimate was calculated for each position estimate. Ideally, the average NEES will be three to represent the three degrees of freedom in the location estimate. The IPA is calculated according to (24) where the TNE of the original NLO and the modified NLOs is compared. TNE and NEES were metrics used in Sprang [1]. The IPA shows areas where the new algorithms are affective and where there are problems when investigating different simulation parameters. The IPA has a range of negative infinity to one hundred where one hundred represents zero error from the new algorithm. If the IPA is positive then it improved the NLO performance while it decreased performance if the IPA is negative.

Many of the variables which are randomly changed are simple to analyze. Measurements per sensor, number of sensors, sensor error parameters and such directly translate in a meaningful way; however, it is a little more difficult to determine how much more complicated one path is compared to another. The simple paths are easy to define because they are completely defined by the altitude of the orbit. The complex paths are created by changing an orbital body with a changing applied force. In a similar fashion to the split and merge modification, the complexity will be translated in terms of the kinematic model used by the NLO algorithm. For each estimate point, the NLO estimates the most likely kinematic model with a constant acceleration. The more complicated a path is, the more the acceleration will change throughout the path. Path complexity is expressed by the variance of the path acceleration as shown in

$$Complexity = (Var[\ddot{x}] + Var[\ddot{y}] + Var[\ddot{z}])/3 \tag{25}$$

by computing the variance of each of the acceleration elements.

The NEES shown in (23) is calculated as it was by Sprang; however, there are

dangerous elements in analyzing NEES values that require a slightly altered method. Given the non-linear nature of the algorithm being tested as well as some higher order elements present in the kinematic model estimation, divergence and lower probability cases will create TNE and NEES values that will be outliers to the distribution of the convergent cases. These outliers can heavily affect the ANEES. The concern of these low probability cases while using the ANEES metric is further described by Li. Li states that using the ANEES heavily penalizes optimism and can be heavily affected by a one or a few large values [16]. In order to mitigate these problems with the ANEES, before estimating the degrees of freedom from the Monte-Carlo simulation, the variance of the NEES vector from each path will be noted. If the NLO does not converge or fundamentally fails, the variance of the NEES will be fairly high. Any simulation with a NEES variance outside of three standard deviations of the rest of the NEES variances will be ignored. Once this is done, the NEES values from the remaining trials will be averaged to find the degrees of freedom. In order for the average NEES to be a valid metric of the degrees of freedom and reflection of the algorithm confidence accuracy, the NEES must be distributed in a chi-squared distribution. This is tested by applying a goodness of fit test on the resulting histogram of the NEES values as well as analyzing the statistical metrics of the NEES distribution created by the Monte-Carlo simulations. The Σ_X^{-2} used to create the NEES calculation will be the position covariance matrix produced by the NLO algorithm Jacobian.

AVNEES.

For the algorithms developed in this thesis, the position confidence is not just a function of the position covariance matrix but the velocity and acceleration covariance as well. Given a large amount of variance on these higher order kinematic elements, the

confidence eclipses should grow even if the position covariance is small. With the higher order elements involved in the position estimation, the position estimation error can be described as

$$PositionError = \mathcal{N}(0, \Sigma_{position}) + time\mathcal{N}(0, \Sigma_{velocity}) + \frac{1}{2}time^2\mathcal{N}(0, \Sigma_{acceleration}),$$

$$PositionError = \mathcal{N}(0, \Sigma_{position} + time^2\Sigma_{velocity} + (\frac{1}{2}time^2)^2\Sigma_{acceleration}). \quad (26)$$

$$\Sigma_X^2 = \Sigma_{position} + time^2\Sigma_{velocity} + (\frac{1}{2}time^2)^2\Sigma_{acceleration}. \quad (27)$$

assuming unbiased estimates where time is the length of the measurement window. While the NEES Σ_X^{-2} just uses the position covariance matrix, the AVNEES will use the position, velocity and acceleration covariance matrices to create position confidence estimates as shown in Equation (27).

This new confidence Covariance matrix attempts to remove higher order elements from the estimate to create a more accurately Gaussian metric. If the velocity and acceleration covariances are large enough with varying time windows, the $time^2$ element could create non-Gaussian error distributions which would leave the position NEES estimate statistically inaccurate. If the $time^2$ element is the only higher order element that could potentially create a non-Gaussian NLO error distribution, the AVNEES should successfully create confidence ellipses that are Gaussian. The AVNEES does not take into account how well the NLO converges or how much each position estimate is affected by prior knowledge created from the rest of the path as well as the affect of an unknown bias. Initial MATLAB simulations have shown that this metric is more affected by an estimation bias than the normal position NEES and could be a problem for the

KMAVNLOCI or cases where the BDNLO and BDMOD methods do not estimate the sensor Bias-Drifts well.

Variable Variation.

In a controlled environment, the effect of one variable on the NLO algorithm can be tested. There are many variables that can potentially effect performance to include: Number of Sensors (NS), Measurements Per Sensor (MPS), complexity of the trajectory, trajectory type, amount of sensor jitter, amount of sensor bias, amount of sensor drift, and many more. By making each of these variables a uniformly random element across a reasonable range, performance metrics can be shown against one or more random variables at a time to find trends. By observing the on average effect of the random variables on algorithm performance described in the previous section, it will be known which variables will have a more significant effect on performance and discover the upward limits of the new NLO algorithm. Once the important variables are discovered, a more controlled simulation can be done to more accurately characterize the relationship in between object tracking conditions and NLO performance.

Another type of variable variation is done where more complex problems will be simulated. In some tests, the sensors are given differing sampling frequencies to see how the algorithm responds to having much more information from one sensor and little from another. In another test, sensor feeds are randomly on and off in order to see how the NLO responds to spotty sensor coverage. This will show how well the NLO can fill in gaps of information by optimizing over an entire path. Similarly, once particular limits are discovered through this initial broad testing approach, a more specific Monte Carlo sweep test is used. The variable variation test described here in essence creates a probability function. Given a reasonable range of parameters, the PDF of performance is estimated by

a large number of tests. The results from those tests form histograms which characterize the probability of performance metrics in relation to the randomly generated simulation variables.

Sweep Tests.

In the sweep tests, everything is held constant except one parameter or two which will be incremented across a range. These tests will be designed based on the results of the variable variation tests. For instance, if the measurements per sensor is found to be an important factor, a set of similar paths and sensor configurations would be set constant and then the measurements per sensor would be incremented from a very low number to a high number and the NLO algorithms would be applied to each increment. If the NLO breaks down when there is a gap in between many measurements, everything would be held constant over a set of paths except the length of the information gap in time until the NLO fails. The purpose of these tests will be to discover and fully characterize the limits of each NLO algorithm.

3.8 Conclusion

These new NLO algorithms have potential to significantly improve Sprang and Hartzell's AOA NLO algorithm. The current NLO algorithm can create accurate estimates on simple trajectories. The trajectory segmentation method could allow the NLO algorithm to handle complicated trajectories where it currently produces less accurate estimates. As well, the bias and drift estimation can potentially transform the error in the LOS data into a zero mean Additive White Gaussian Noise (AWGN) which would be ideal for NLO estimates. These two methods are attempting to transform any type of

trajectory and set of LOS measurements into a data set best assessed by the NLO algorithm. The next chapter shows the results from the variable variation and Monte Carlo tests on these two methods.

IV. Results

4.1 Introduction

In this chapter, the results from the previously described variable variation tests and Monte-Carlo trials are examined and discussed. First the important results from the variable variation tests will be presented highlighting the variables that highly affect the performance of each NLO algorithm and in what way. The general trends discovered through the variable variation tests are also discussed for each of the NLO algorithms. The BDNLO and BDMOD appears to have improved performance of the NLO algorithm. This chapter defines the limits and amount of improvement from these new Bias-Drift estimating algorithms. The SMNLO algorithm did not appear to affect the performance in a positive or negative way and will be excluded from this chapter. Results from the SMNLO can be seen in Appendix A. For reference the algorithms concise description can be seen in table 1.

Table 1. Algorithms.

Algorithms	
KMAVNLOCI	Kinematic model constant acceleration estimates with no bias drift estimate
SMNLO	KMAVNLOCI with complicated paths broken into smaller sections
BDNLO	KMAVNLOCI with extended jacobian to estimate sensor bias and drift
BDMod	Uses KMAVNLOCI as a module while correcting sensor bias drift separately

First, the results from the tests to discover when AOA localizations are in fact multivariate Gaussian will be presented. Next the results from the variable variation tests are presented, highlighting the algorithm general trends and significant variables for each algorithm. The variable variation test discusses the over all performance affect from the Bias-Drift estimating modifications on specifically the KMAVNLOCI algorithm. The

variable variation tests are randomly generated uniformly using a reasonable range of values. These ranges are further described as the test results are presented. In the variable variation tests, the IPA metric is used to describe the performance improvement and the NEES metric will be used to describe the affect on confidence. Lastly, a set of unique simulation conditions will be created to find hard limits of all of the NLO algorithms in the controlled Monte Carlo Simulation (MCS) or sweep tests. In the sweep tests, everything will be controlled except one or two parameters and the algorithm will be applied to a set range of those parameters. Those parameters include path type, number of sensors, measurements per sensor, sensor sampling frequency, gaps in measurements and different magnitudes of the different error parameters.

4.2 Covariance Intersection Validation Tests

Two sensors were set at coordinates $(0, 0)$ and $(0, dist)$ where $dist$ or distance would change depending on the parallax measurement being tested. The object being sensed is located at $(5, dist/2)$ where the units are some arbitrary distance unit. Gaussian noise is added to a set measurement angle and the resulting localization is made for that noisy measurement. The resulting distribution is found to change drastically depending on the variance of the AWGN and the parallax angle. Small parallax angles and parallax angles close to 180 degrees are more likely to create non-Gaussian localization distributions under much lower measurement variance values than other parallax angles. 100,000 measurements with a Gaussian measurement variance of 0.0001 radians was simulated with a parallax angle of 0.0115 degrees. Figure 14 shows the 2D histogram of this simulation. The same simulation was done with a variance of 0.005 radians and a parallax angle of 177.7085 degrees. Figure 15 shows the 2D histogram of the resulting estimates. It is interesting to note that both the small and large angled parallaxes create

estimates that appear to be biased. While the object being triangulated is at a Y value of 5, in the shown search area the small angle simulation contains an average estimate Y value of 4.2258 while the large angle simulation contains an average estimate Y value of 4.0015. There are as well plenty of estimates not shown in the graph very far away from the sensors in the small parallax angle scenario, when those are accounted for there is a new average Y value estimate of 12.51 for the small parallax angle test.

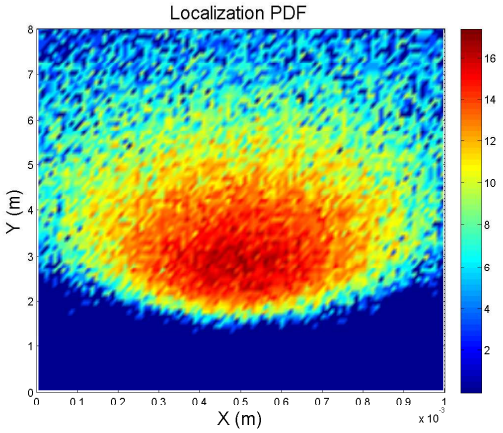

Figure 14. 2D Histogram of small parallax localizations

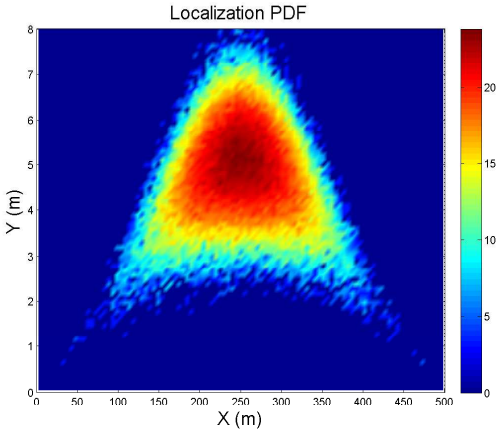

Figure 15. 2D Histogram of large parallax localizations

The simulations shown in Figures 14 and 15 use a bad sensor configuration and quickly create non-Gaussian localization distributions as variance increases. On the other hand the simulation of a good sensor configuration with a parallax angle of 90 degrees can as well be turned into a non-Gaussian localization distribution at higher measurement variance values. As seen in the 2D histogram in Figure 16 a high enough variance will reveal that no matter what the parallax angle the sensor configuration creates, the localization distribution is not necessarily Gaussian. However, Figure 16 was created with a measurement variance of 0.1 radians which is unreasonably high for most AOA sensors. Using the Royston's test made by Trujillo-Oritz, an angle-variance sweep was simulated to discover the parallax-angle variance limit where localizations became a non-Gaussian distribution [6]. Figure 17 shows the resulting limit where any variance and angle of

intersection below the line can be safely assumed Gaussian. The angles close to 0 and 180 degrees require extremely low variance while many of the angles in between are well above the amount of variance expected in space based systems. As long as the satellites are separated by at least a few degrees in relation to the object being sensed, the localization PDF will in essence be Gaussian and CI is justified in a very simple scenario. The localization error distribution can still end up not being Gaussian based on other factors in the NLO algorithm. Testing this hypothesis is done in the discussion of the variable variation NEES.

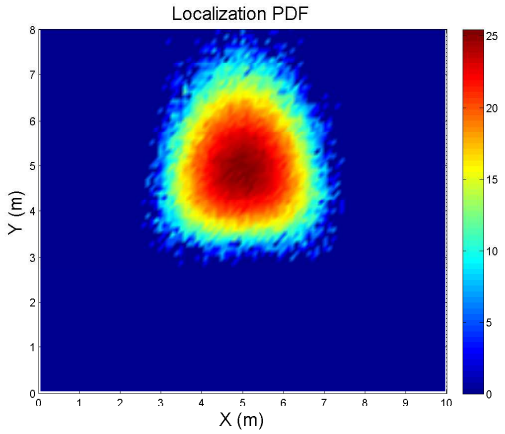

Figure 16. 2D Histogram of right angle parallax localizations

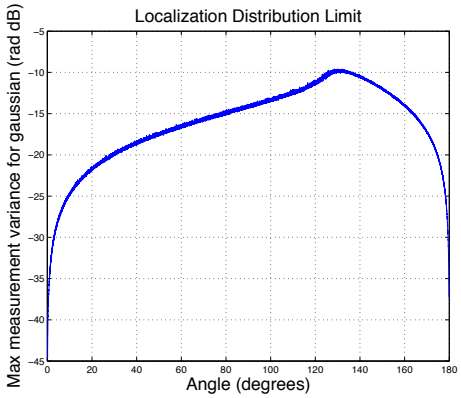

Figure 17. Maximum variance before AOA localization is no longer Gaussian

4.3 Variable Variation Tests

This section describes the PDF of IPA and NEES performance resulting from the variable variation tests. First the IPA results from the BDMOD tests, then IPA from the BDNLO tests, then the NEES of both the BDMOD and the BDNLO in comparison to the KMAVNLOCI, and then finally the effect of random visibility on each NLO algorithm. In general, the simple path trajectories can be estimated regularly and as predicted in Chapter III. As soon as a changing force vector is applied to the simulated trajectories, there are many problems that arise. The irregular and ever changing acceleration affects the ability

of the constant acceleration model to converge to a stable estimate. While there is a fairly small affect on the more stable KMAVNLOCI and BDMOD algorithms from the complicated paths, the BDNLO was found to be highly affected by the change in path type.

Bias Drift Module.

The initial variable variation tests incorporates a random configuration of satellite based sensors uniformly distributed between 3 and 8 sensors. Each trial uses a different MPS value which would be uniformly random in between 200 and 800 measurements. The first variable variation tests contain the same MPS for each sensor. Different MPS values for each sensor within the same trial is tested as well but separately. Each sensor was given a trajectory in a simulated random orbit with a constant direct LOS with the object being tracked; this limited the sensors to a same side of the Earth as the object. In the simple path case each object being tracked was given an altitude and an appropriate orbit. In the complicated path case an object was given an initial orbital or sub orbital position and then given a changing and random force vector. Orbital mechanics were then applied and a complicated path could be simulated.

The BDMOD behaved as expected by converging more often than the BDNLO method, but in ideal conditions produced less accurate estimates than the BDNLO. The BDMOD converged 98.6% of the time and improved the geolocalization estimate 95% of the time in the simple case. The BDMOD averaged a IPA of 20.8% for the simple path. In the complicated path tests the algorithm converged at about the same rate of 98.2%, but with a decreased improvement rate of 86.3% of the time. The average IPA in the complex case is about half to 10.9% for complicated paths. The IPA histograms of the BDMOD for these two tests can be seen in Figures 18 and 19. It appears that there is a low risk

involved with the BDMOD given that the Bias-Drift is not perfectly estimated, but still converges and is expected to improve performance. The worst result from the BDMOD is a small decrease in accuracy less than 20%. The risk seems to be similar for the complicated paths as well.

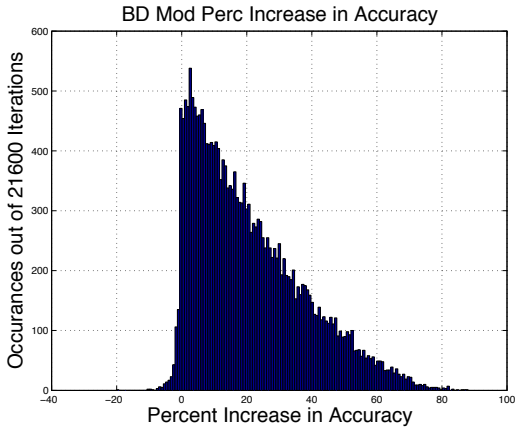

Figure 18. BD Mod Histogram for a simple path compared to the KMAVNLOCI.

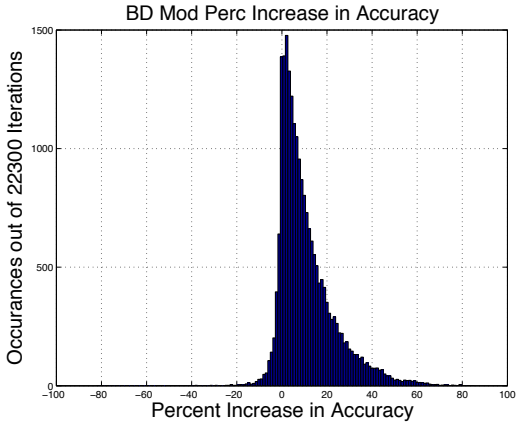

Figure 19. BD Mod Histogram for a complicated path compared to the KMAVNLOCI.

As shown in Figures 22 and 23, the more sensors used in the simulation the higher the chance is of a large improvement in accuracy. Having more sensors does improve the accuracy of any sort of geolocalization algorithm in general so this is a common trend; however, Figures Figures 22 and 23 show IPA which expresses how much more accurate the BDMOD makes the KMAVNLOCI. When the number of sensors used is lower, the improvement is smaller and less significant. As more sensors are used better bias-drift estimates can be made and there is more significant improvement to the geolocalization estimate. In Figure 22 there is much more of an upward improvement trend than in Figure 23. There is still more variance in the positive direction, but not as much. This could be the result of less accurate initial position guesses from the KMAVNLOCI.

When the affectiveness of the BDMOD is shown against the TNE of the KMAVNLOCI, there is a consistent upward trend for the simple and complicated cases as seen in Figures 22 and 23. The KMAVNLOCI is the algorithm used for comparison and

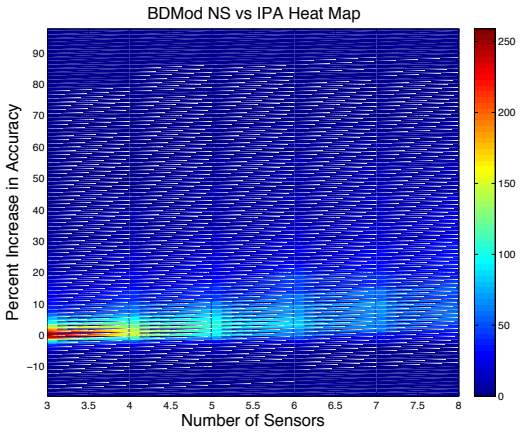

Figure 20. KMAVNLOCI IPA vs NS BD Mod 2D Histogram for a simple path.

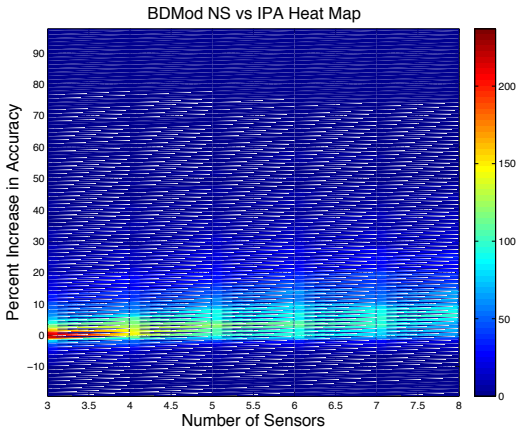

Figure 21. KMAVNLOCI IPA vs NS BD Mod 2D Histogram for a complicated path.

the initial bias-drift guess for the BDMOD. The new bias-drift estimations seem to create a positive feedback loop which increases accuracy when the KMAVNLOCI performs well. As the NLO module performs worse, the BDMOD affect becomes more and more insignificant. The BDMOD improves the NLO on average, but it can't fix a bad initial estimate. It is important to note that the random inputs were sensing conditions and the hottest spots seen is from the most likely amount of error from KMAVNLOCI given that particular type of test. This is why Figures 22 and 23 do not show a uniformly distributed TNE for the KMAVNLOCI.

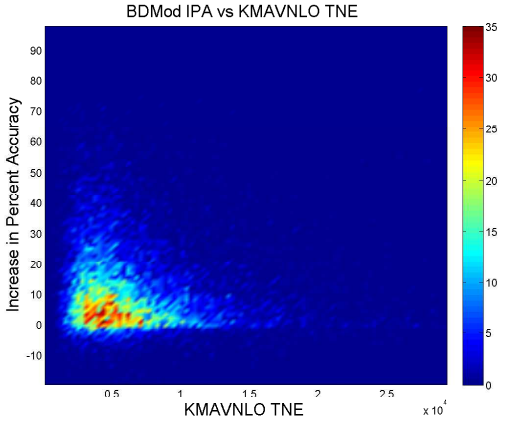

Figure 22. KMAVNLOCI TNE vs IPA BD Mod 2D Histogram for a simple path.

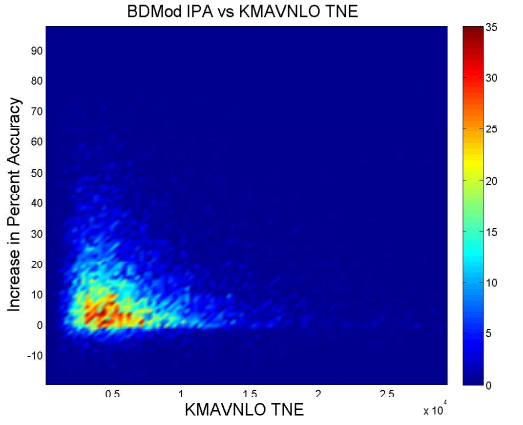

Figure 23. KMAVNLOCI TNE vs IPA BD Mod 2D Histogram for a complicated path.

Bias Drift NLO.

The BDNLO algorithm was tested under the same conditions as the BDMOD to produce the following results. As expected, the BDNLO produces a much lager accuracy improvement than the BDMOD but is more likely to diverge. In the simple path tests the BDNLO converged 97.5% of the time with a 93.7% improvement rate, and an average IPA of 68.3% given convergence. In the complicated path tests the BDNLO converged 92.3% of the time with a 69.17% improvement rate, and an average IPA of 22.87% given convergence. The overall histograms of the BDNLO IPA can be seen in Figures 24 and 25. There is much more accuracy reward in the BDNLO than the BDMOD. There is much more risk involved in the BDNLO. Even though the mean improvement is much higher for the BDNLO than the BDMOD, the BDNLO left tail is much longer in both the complicated and the simple path tests. This means that if the BDNLO fails, there will be much more loss in accuracy. On the other hand, especially in the simple test, the BDNLO frequently creates extremely accurate geolocalization estimates. In these ideal cases, the BDNLO is the superior algorithm. The complicated path drastically increases the risk of the BDNLO. Even with the much higher risk, the BDNLO will still on average beat the BDMOD in the complicated path tests.

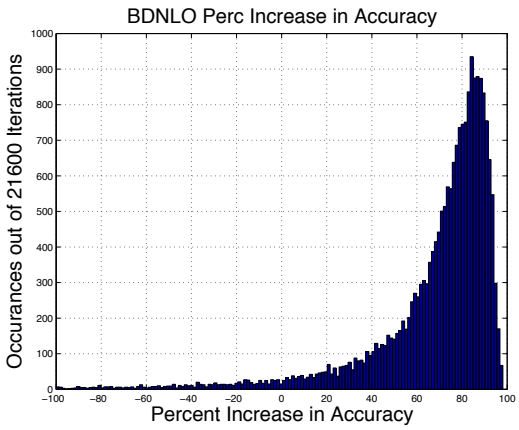

Figure 24. BDNLO Histogram for a simple path compared to the KMAVNLOCI.

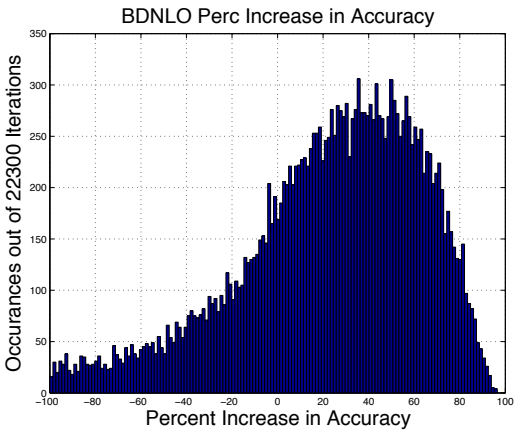

Figure 25. BDNLO Histogram for a complicated path compared to the KMAVNLOCI.

The BDNLO method responds to the number of sensors in a very different way. The histogram of BDNLO performance in relation to the number of sensors can be seen in Figures 26 and 27. In the simple path test there is not much affect from the sensors, other than a tighter distribution at 5-6 sensors or higher and a wider more riskier distribution at a low number of sensors. In the complicated path tests there is a steady decline in BDNLO accuracy improvement as more sensors are added to the estimate. While this may be counter-intuitive, it makes sense in the context of the BDNLO's source of instability. The BDNLO creates a larger Jacobian by adding $4 \times Numberof Sensor$ columns. The larger Jacobian becomes the harder $J^T J$ is to invert creating an even more unstable NLO iterative step. This problem will disappear as this algorithm is applied to more precise mathematical methods; however, the simulations presented in this thesis have been run in MATLAB where the machine precision can accurately invert matrices with condition numbers up to 10^{14} before precision based error becomes a factor. In the variable variation tests, the BDNLO can easily produce a $J^T J$ with a condition number as high as 10^{25}. On the other hand, the decline in accuracy improvement could be the result of the KMAVNLOCI improving more than the BDNLO for higher number of sensors on a complicated path. This is investigated much closely in the more controlled sweep tests.

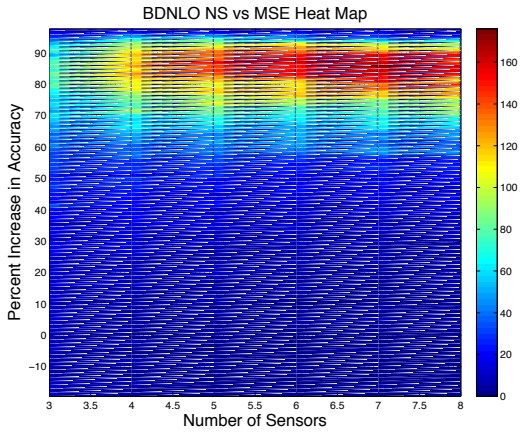

Figure 26. KMAVNLOCI IPA vs NS BDNLO 2D histogram for a simple path.

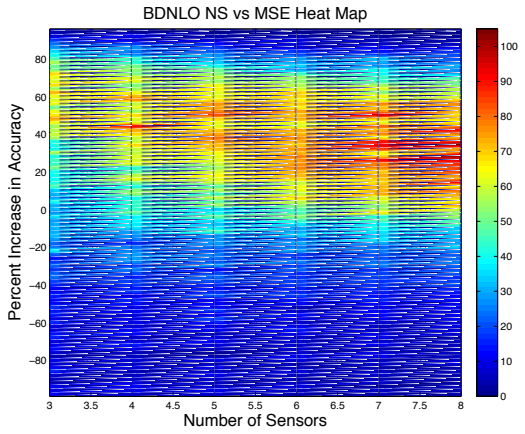

Figure 27. KMAVNLOCI IPA vs NS BDNLO 2D histogram for a complicated path.

While being more unstable than the BDMOD, the BDNLO also contains a very different relationship with the performance of the KMAVNLOCI than the BDMOD method. These relationships can be seen in Figures 28 and 29. In both tests the BDNLO has the opposite relationship with KMAVNLOCI than the BDMOD. The BDNLO increases the accuracy on average much more when the KMAVNLOCI TNE is lower, and less often when the the KMAVNLOCI TNE is higher. The BDNLO maintains this relationship for the simple and complicated paths. In general, this is because there is much more for the BDNLO to improve. The BDNLO improves the accuracy independent of how the KMAVNLOCI performs. This is great for cases where the Bias-Drift error in a system significantly decreases the accuracy, because with the BDNLO it does not necessarily matter if the initial estimate is accurate. Like the other BDNLO results it is unfortunate how sensitive the BDNLO is in the complicated path tests. As seen in Figure 28 it is extremely unstable.

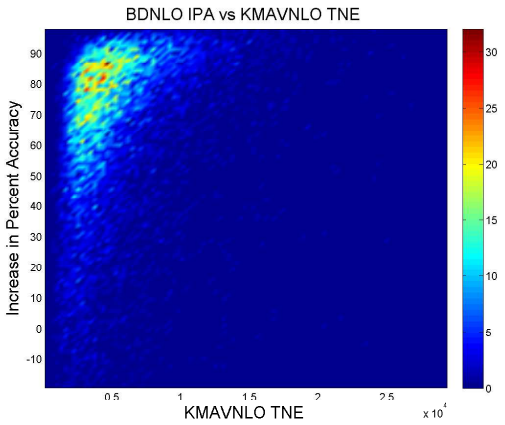

Figure 28. KMAVNLOCI TNE vs IPA BDNLO 2D Histogram for a simple path.

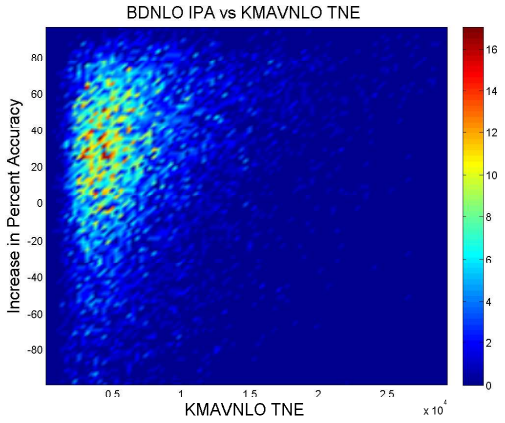

Figure 29. KMAVNLOCI TNE vs IPA BDNLO 2D Histogram for a complicated path.

Confidence Credibility Analysis.

In the variable variation test the NEES was recorded as well as the AVNEES for each position estimate, trial and algorithm. Ideally, the NEESs would be distributed by a

chi-squared distribution with three degrees of freedom because the NEESs includes the three ECEF location elements which would ideally be normally distributed and normalized by the NEESs. The resulting NEES values did not appear to be chi-squared with three degrees of freedom. As presented and discussed in Appendix C, the distribution is not quite a chi-squared but close enough for the NEES to be a valuable metric. The AVNEES appears to be closer to a chi-squared distribution than the NEES, but there still appear to be non-gaussian elements unaccounted for in the confidence metrics. For each of the tests the divergent cases were not included in the NEES analysis because the few large NEES values effectively became outliers driving the mean much higher than the true mean of the convergent cases. As expressed in [16], outliers more easily affect ANEES more significantly than many other metrics. The resulting NEES CDFs of the variable variation tests can be seen in Figures 30-33. The CDFs appear to be much more under confident in the simple paths and express much longer right tails when used on complicated paths. Many of the NEES and AVNEES histograms can be seen in Appendix C which reflect the longer tails as well. While the AVNEES diminishes the right tail of the complicated path NEES significantly, it creates even more under confident estimation estimates for the simple paths. Similarly to the previous sections in this chapter, the simple path tests perform much better than the complicated path tests. The closer the NEES distribution is to a chi-squared distribution with three degrees of freedom, the better the resulting error ellipses from the CI is for modeling and estimating the NLO position confidence.

Important metrics from the NEES of each algorithm can be seen in Table 2. For reference, an ideal chi-squared of three degrees of freedom will have a mean of 3, a median of 2.37 and 5% of the distribution above 7.82. For each of the algorithms, the simple path created NEES values that reflected an under confident estimation. The AVNEES values were even more under confident. On the other hand, the complicated paths on average created NEES values that reflected over confident estimations. While the

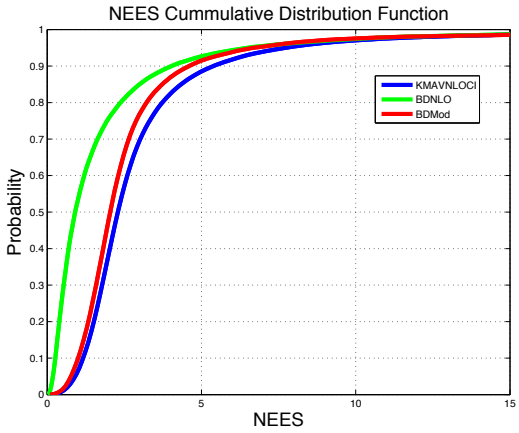

Figure 30. NEES CDF for a simple path.

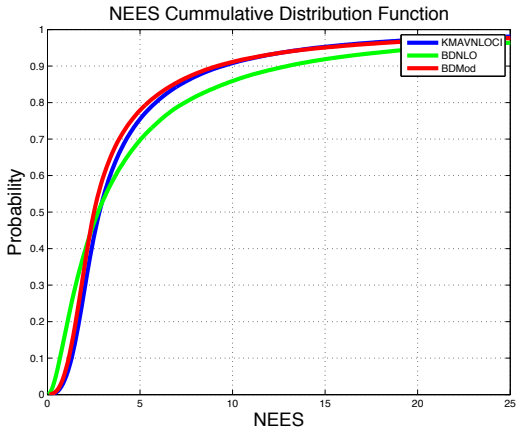

Figure 31. NEES CDF for a complicated path.

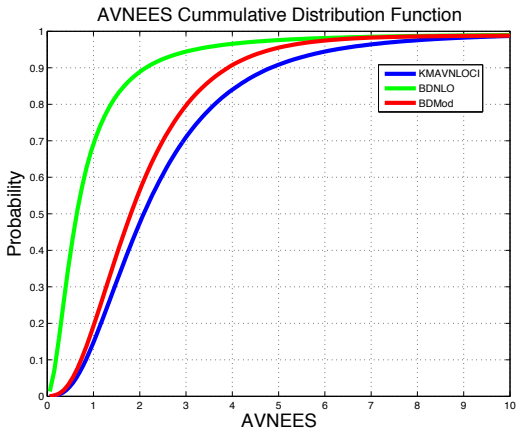

Figure 32. AVNEES CDF for a simple path.

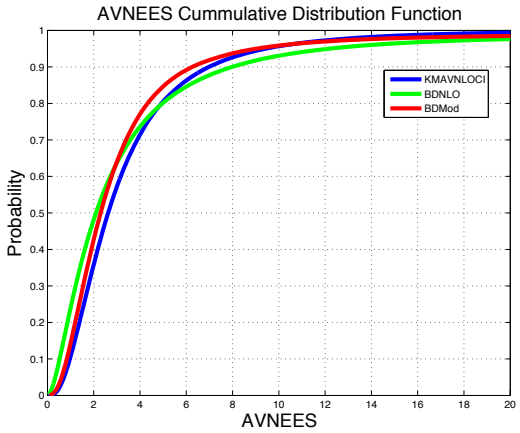

Figure 33. AVNEES CDF for a complicated path.

AVNEES values were more under confident for the simple path, the new metric improved the performance during a complicated path estimation significantly. The complicated paths created many more unstable systems with much less accurate localization estimates. All of the NEES statistics increased in some way in the complicated path tests. Once the acceleration and velocity variance became an integral part of the position estimate, a more unpredictable kinematic model could not affect the confidence metric as much as it would when acceleration and velocity variance is ignored. As seen in the histograms in Appendix C, all of the complicated path NEES PDFs contain larger right tails. This will affect the mean and the 5% confidence interval much more than the median which is

reflected in Table 2. As expected from the previous results, the BDNLO is more drastically affected by the complicated path in NEES as well as accuracy and convergence rate. When convergence and accuracy is more likely, the BDNLO is the most under confident, otherwise the BDNLO is extremely overconfident. This affect is seen in the other two algorithms but not nearly as significantly as the BDNLO.

Table 2. NEES metrics.

NEES	Simple Path		
	5% Confidence	Mean	Median
KMAVNLOCI	7.72	3.33	2.29
BDNLO	6.50	2.04	0.906
BDMod	6.67	3.34	2.04
NEES	Complicated Path		
KMAVNLOCI	14.52	5.08	2.81
BDNLO	20.71	6.17	2.73
BDMod	14.86	5.09	2.52
AVNEES	Simple Path		
	5% Confidence	Mean	Median
KMAVNLOCI	6.23	2.70	2.09
BDNLO	3.20	1.73	0.64
BDMod	4.84	2.70	1.81
AVNEES	Complicated Path		
KMAVNLOCI	9.44	3.70	2.62
BDNLO	12.2	3.61	2.10
BDMod	9.04	3.70	2.30

Each locational value was saved during a simple path variable variation test as previously described. The histogram of the Euclidean x, y and z error can be seen in Figure 34 fit to a Gaussian distribution. The error probability would easily be Gaussian given smaller tails for each of the algorithm. As each NLO algorithm becomes more accurate, the center spike grows more probable, but the tails contain the same effect as seen in the BDNLO histograms. What causes this non-Gaussian distribution is the

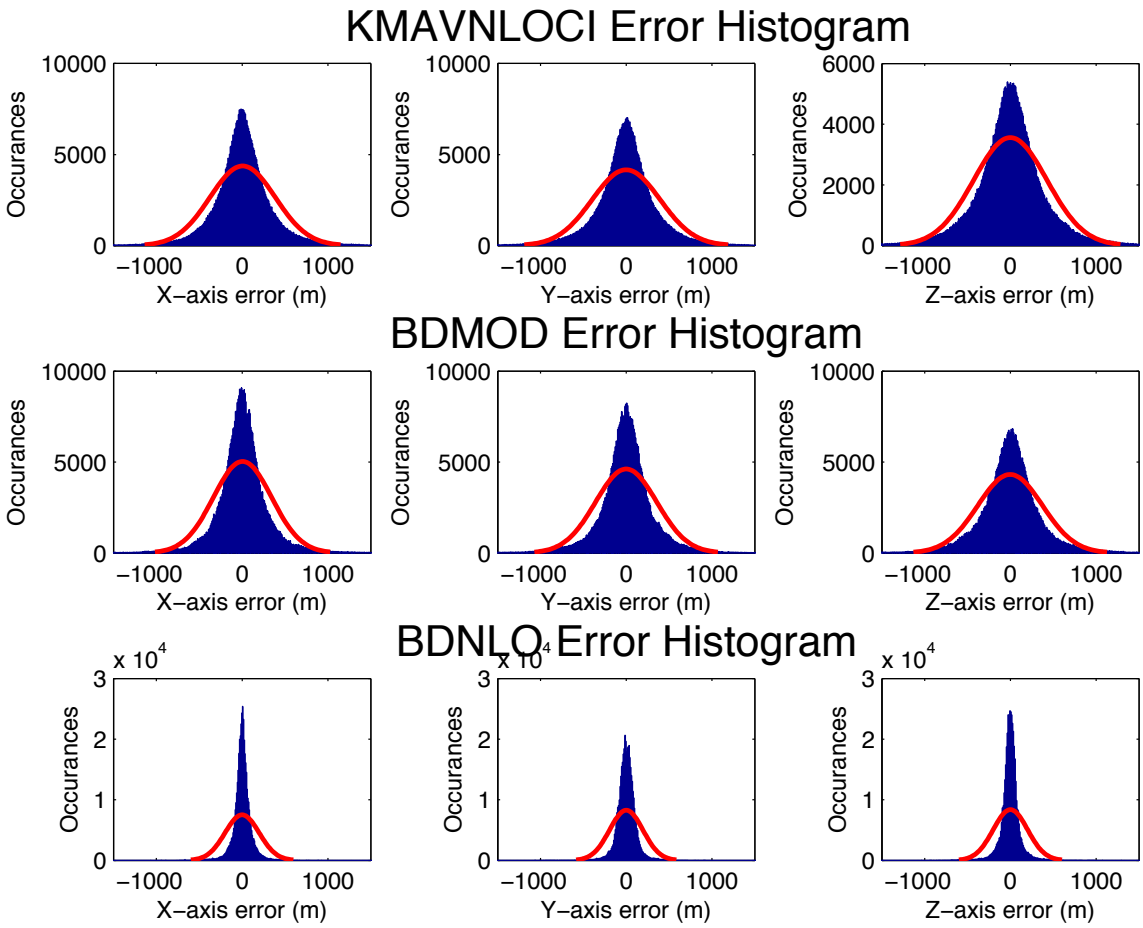

Figure 34. Locational error histogram with fitted Gaussian curve.

accuracy of the kinematic model as well as the varying time period of the estimation windows. Accuracy of the kinematic model elements for each NLO trial such as velocity and acceleration estimates cause the location estimates to have a second order error function in relation to a random windowed time period. On a trial by trial level, the locational error was found to be similar to a smaller trajectory. This suggests that the error could also be characterized as a kinematic error over the optimized path estimate and not just for each individual position estimate. The affect this distribution function has on the NEES values is discussed in depth in Appendix C.

Visibility Variation Test.

In this test a normal variable variation was run under similar conditions as before, but with a random selection of active sensors for different time blocks. The average visibility across the path as well as the visibility variance and lowest sensor visibility was recorded for each trial. As seen in Figure 35 for a simple path under random visibility, there is a much higher chance of divergence, but when the algorithm converges the NLO algorithms will perform as it normally would have under more normal conditions. While the algorithm would normally converge at a rate generally over 90%, the random visibility creates a convergence rate around 73% for each of the NLO algorithms tested. Figure 35 also shows that the BDNLO is much more affected by sensor visibility than the KMAVNLOCI and the BDMOD. The BDNLO fails more spectacularly than the other algorithms as well but that does not matter when the failure is known.

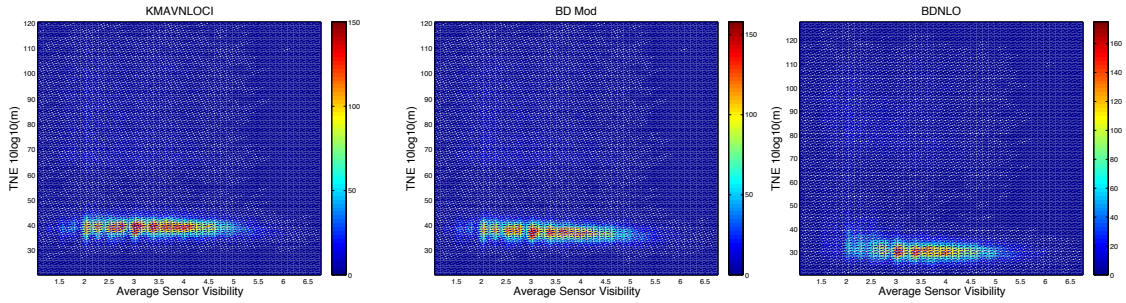

Figure 35. Sensor visibility, simple path.

This test was done on a set of complex paths as well. There were no significant changes in the relationship between visibility and path type other than the normal decrease in accuracy and a slight increase in the divergence rate which has been seen before. The results of the complex test can be seen in Apendix E. The next step in this analysis is to determine how large gaps in sensor measurements can be before the NLO algorithms begin to break down. This is tested in a sweep test which increments the size of a time gap in information for a set path and sensing conditions until the NLO begins to diverge.

4.4 Sweep Tests

In the more controlled sweep tests, the affect of different numbers of sensors and measurements per sensor will be tested together, changing sensor sampling frequencies, gaps of measurements, and different error model parameters will be tested along a range while keeping all other parameters constant. These tests reveal when the selected parameters become significant and could be expected to produce much less accurate estimates or even a larger chance of divergence. These tests more importantly reveal room for improvement in the NLO algorithms for many possible but unforseen scenarios.

Number of Sensors vs Measurements Per Sensor.

A sweep test was done on twenty set paths across a grid of sensor parameters including number of sensors and measurements per sensor. The test was done on a simple and complex path ranging from two to ten sensors and one hundred to one thousand measurements per sensor. Figures 36 and 37 show the results from this test. In the simple path case shown in Figure 36, the KMAVNLOCI and the BDMOD perform regularly across number of measurements with a gradual increase in accuracy as the number of sensors increase. There appears to be some trouble in the bottom left corner where there are not many measurements or sensors. The BDNLO performs in a similar but more extreme way. There is little chance of convergence at two sensors and there is a sharp increase in accuracy as the number of sensors is increased. There is still the trouble area in the bottom left corner where there are not many sensors or measurements. For the BDNLO as soon as regular convergence is achieved, accuracy is much higher than the other two NLO algorithms on average. The complicated path tests shown Figure 37 more prominently shows the problems that appear to form in the simple path test. There is much

more divergence on the left, low number of sensors, in all of the NLO algorithms especially the lower left corner with a lower amount of measurements. The BDMOD algorithm begins to have problems with more divergent cases as shown by the random hot spots that were not in the KMAVNLOCI case. It is important to note that the error in this test was averaged in normal space and then tranformed into log space to display the divergent and convergent cases side by side, the hot spot is one or two divergent cases that drove up the mean in the BDMOD result. In the BDNLO, the lower left corner divergence zone became more prominent and the over all accuracy decreased as seen by the upper right portion of the figure. The larger error at low number of measurements at a high number of sensors would explain the decrease in IPA seen earlier in the variable variation tests seen in Figure 27. The difference between the complex and the simple case can be attributed to the less regular acceleration of the object being tracked. This gives the NLO a much harder time estimating a single acceleration for all of the windowed points. For a lower measurement per sensor the BDNLO could easily have problems converging on the bias and drift of a sensor just in a simple case. Stability decreases even more once the path contains an irregular acceleration.

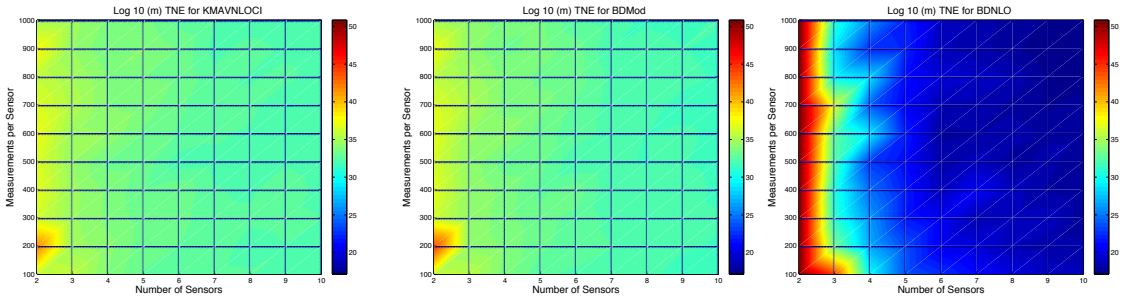

Figure 36. NS vs MPS TNE, simple path.

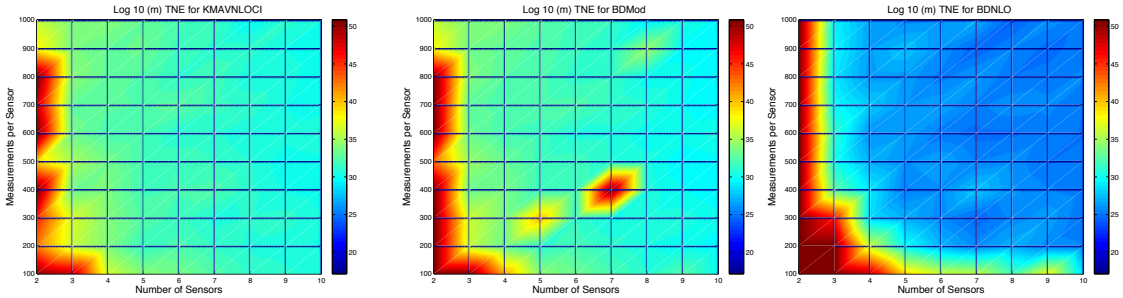

Figure 37. NS vs MPS TNE, complex path.

Sensor Sampling Frequency Proportions Test.

A sweep test was done to characterize the behavior of the NLO algorithms in a mixed sampling rate scenario. Over five different simulated simple and complicated path types, sensor configurations and error parameters, the BDNLO, BDMOD and the KMAVNLOCI was tested on different measurement concentrations. A set amount of total measurements were used for each trial, but the percentage of those measurements was incremented from a even distribution to a highly unbalanced one. As the measurement distribution became more and more unbalanced, a dominant sensor would contain a growing percentage of the total measurements and the other sensors would evenly split the remaining measurements. Figure 38 shows the average error of the 5 trajectory localizations with three sensors on a simple path. It is clear to see that in this test, as the sensor measurements became more unbalanced, the NLO algorithms were more likely to fail, and when around 85% of the measurements were coming from the dominant sensor, the NLO algorithms began failing regularly. Across the complicated path and different total number of sensors, the algorithms all began to fail under the same unbalanced measurements conditions. The remaining test results and further discussion can be seen in Appendix D. This could be useful when combining the results from many different types of sensors. Suppose some mixture of old sensors with low sampling frequencies are tracking the same object as one new sensor with a very high sampling frequency. While

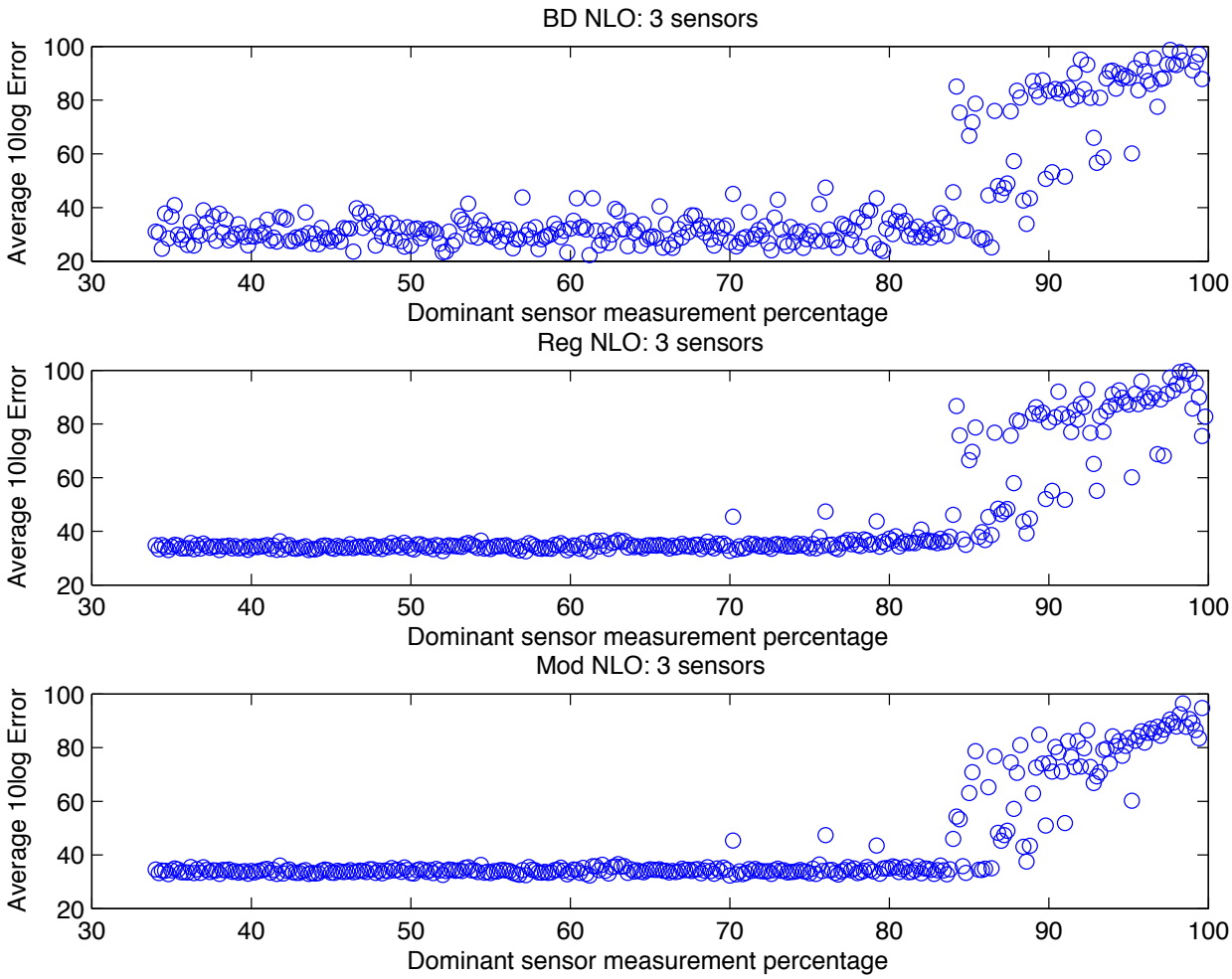

Figure 38. Unbalanced sensor measurement concentration sweep: 3 sensors, complex path.

more information from one sensor will not hurt the performance, not having enough information from the older sensors for each estimate can. This suggests the need for a smarter windowing function that would automatically guarantee enough measurements from different sensors for each location estimate.

Widening Gap Test.

For a simple path and a set of four sensors, the NLO algorithms were tested with an increasing gap of measurements. This was tested in a scenario when one sensor was

tracking the object through the gap and when no sensors were making measurements during the gap. The TNE results can be seen in Figure 39. When there were no measurements in the gap, the algorithm functioned normally. However, when there was only one sensor active in the gap, the NLO algorithms failed around ten seconds. This is in large part due to the measurement windowing for each estimate as well as the mixed measurement density results discovered in section 4.4. During the gap, the algorithms are still attempting to make measurements based on one sensor and only a few measurements from other sensors on the edge of the gap. This problem could be easily fixed by creating a more adaptive windowing method.

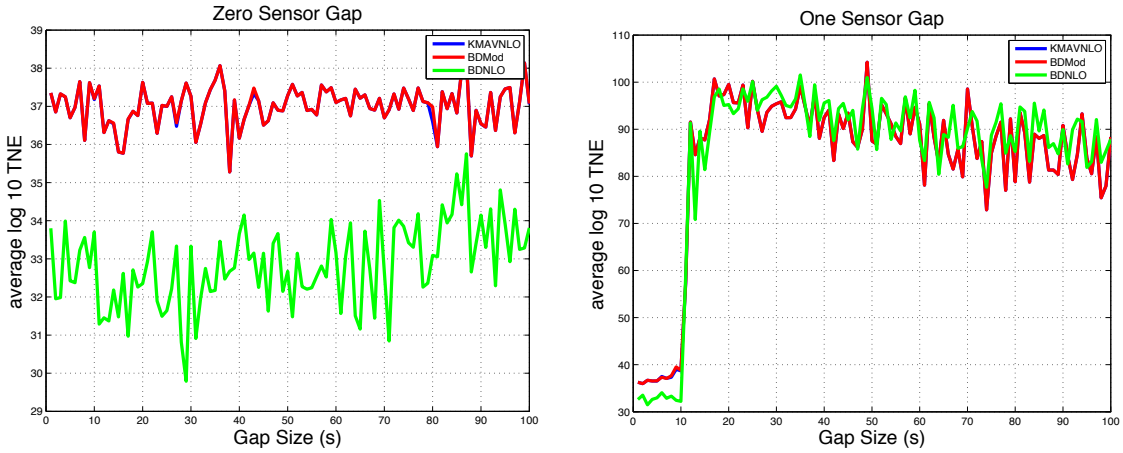

Figure 39. MSE across different sizes of sensing gaps.

Error Model Tests.

In this test, every parameter remained the same except an error parameter. Figure 40 shows the TNE error averaged over 20 paths (not including divergent cases) for a log-space span of a Bias, Drift and Bias Instability range. For each trial, a random error parameter was chosen uniformly distributed up to that Bias-Drift-Bias Instability factor for each sensor. If the error parameter was not changing, it remained at the same value used in previous tests. As seen in the linear bias and drift test results in Figure 40, at a

certain point, the KMAVNLOCI and the BDMOD begin to diverge. The BDMOD tends to improve the KMAVNLOCI more and more as the Bias-Drift elements increase it is still limited to the initial estimate accuracy of the KMAVNLOCI results. On the other hand, the BDNLO is unaffected by the Bias and Drift values within reasonable values. It is able to converge to a regular solution up to very high Bias-Drift values. Before the value of $2 * 10^{-4}$ for the Bias and value of $2 * 10^{-7}$ for the Drift, the KMAVNLOCI performs steadily well as the BDNLO and has much less variance in performance than the BDNLO. The Bias test and the drift test match up as well because, at normal conditions, the Bias factor is set to 10^{-4} and the Drift factor is set to 10^{-7} which reflects to the negligible difference in KMAVNLOCI accuracy with the BDNLO in the Bias test before $2 * 10^{-4}$ and the constant and significant improvement in accuracy in the BDNLO before $2 * 10^{-}6$

This test also revealed the amount of bias and drift that the BDNLO can handle before diverging. At a bias values in the range of $7 * 10^{-2}$ radians and drift values of $2 * 10^{-4}$ radians per second, the BDNLO will begin to regularly diverge. Theoretically, this should not happen. The BDNLO should be able to account for any range of measurement bias and drift; however, the initial guess created for these simulations is based off of the erroneous data. The initial guess was tested at length in Appendix B and was found to be possible but computationally tough given the enormous amount of trajectory and sensor bias-drift possibilities. Given a close guess of the true object trajectory, the BDNLO will still converge no matter what the magnitude of the sensor bias and drift. This is not a very pressing issue given a space based sensing system. $7 * 10^{-2}$ radians of Bias and $2 * 10^{-4}$ radians per second of drift are unreasonably large bias and drift magnitudes for many space based IMUs.

4.5 Conclusion

Each algorithm performs as expected. The KMAVNLOCI performs as it did in Sprang and Hartzell. The BDMOD improves the accuracy but does not converge to the most accurate solution. BDMOD is also heavily dependent on the initial accuracy of the KMAVNLOCI algorithm. The BDNLO creates the most accurate estimates but is most likely to diverge. As well, the BDNLO is highly dependent on error looking a certain way. If the bias instability becomes too high, the BDNLO begins performing much worse than any of the other algorithms. The BDNLO is a great solution to a very specific sensor model and for that linear like time drifting model, the BDNLO shows significant improvement.

In many normal conditions, the NLO algorithms perform very regularly. However, extreme conditions shown in the Measurements per Sensor vs Number of Sensor, visibility, measurement gap and error parameter tests have shown that possible conditions can cause unexpected problems with the NLO algorithms.

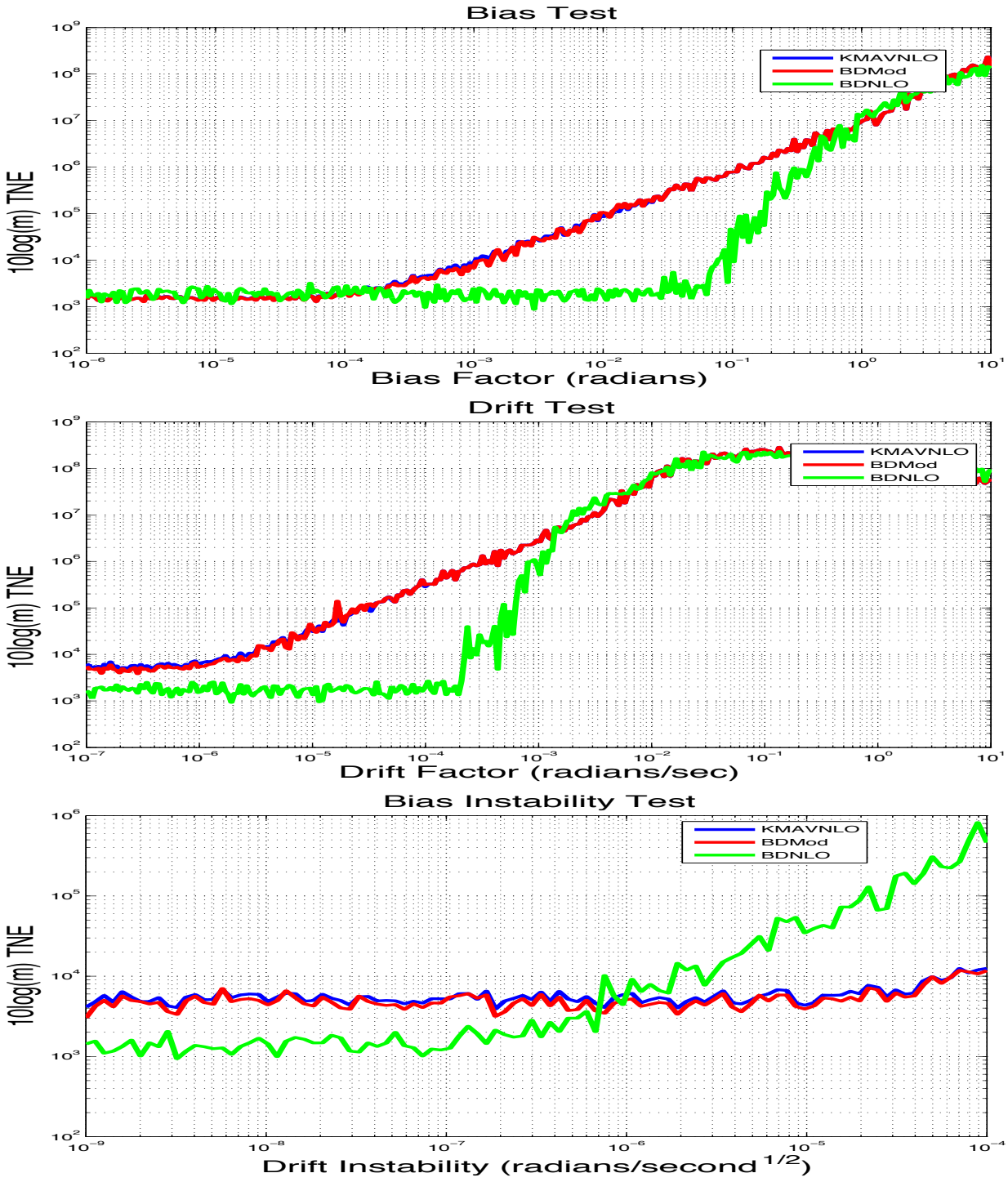

Figure 40. Error Sweep Tests.

V. Conclusion

5.1 Introduction

In this chapter analysis from the results from chapter 4 are discussed and synthesized. Many different simulations have been done and presented for the three NLO algorithms heavily tested in this thesis: the KMAVNLOCI, the BDMOD and the BDNLO. This section will review and further analyze all of these simulations as well as connect the conclusions with the results for Sprang and Hartzell.

The tests in this thesis consisted of two stages. The variable variations tests were first done to find general trends and performance across a large range of random values. Some of the results from these tests turned out to be somewhat counter intuitive, such as when the BDNLO began performing worse as the number of sensors increased on a complex path. To further investigate these problems the second test was designed to be a more depth first approach by holding all elements constant except for one.

The metrics used and discussed were identical to the metrics used by Sprang. This will allow further connections and analysis to be made in between the new algorithms tested and some that were only tested by Sprang to include the Static NLO, VNLO, VNLOCI, and the KMVNLOCI. The new algorithm's NEES and TNE will be compared with the results reported by Sprang.

5.2 Simulations

The simulations began by testing the NLO algorithms randomly within a range of parameters. These tests were split into simple paths representing an object in a simple orbit and a complicated path which included a random force vector. Through these tests

general trends were established which created a basis for a more controlled MCS test which was completed next. The BDMOD proved to be a positive feedback loop for the KMAVNLOCI. As the KMAVNLOCI estimate became more accurate, the BDMOD improved the result even more. Despite the path type the BDMOD behaved in the same manor with a small decrease in accuracy, 20.8% IPA for the simple and 10.9% IPA for the complicated paths. While the BDMOD is expected to increase accuracy regularly, the improvement was marginal compared to the BDNLO which averaged an IPA of 68% for simple paths and 22.87% for complicated paths. The BDNLO improved performance regardless of the accuracy of the KMAVNLOCI. However, as soon as the object path became complicated there was a very large variance in the performance of the BDNLO. The BDNLO performed better than the KMAVNLOCI only 69.2% of the simulation trials. As well, BDNLO was shown to be more affected by random sensor visibility where a low average visibility meant certain divergence.

Next the algorithms were simulated on a set number of paths and with one parameter incremented. In this test the effect of the amount of sensors and measurements, disproportionate sensor sampling frequency, information gaps and error parameter magnitude on the NLO algorithms were tested. In all of the tests, there was a breaking point for each algorithm. The BDNLO tends to diverge whenever there are only two sensors, and more often when there are very few measurements per sensor. The KMAVNLOCI and the BDMOD are affected by low number of sensors but most of the time will still converge. In the sampling frequency test, the algorithms were tested in an environment where one sensor slowly became more and more dominant. Each algorithm began to break down and diverge when the dominant sensor contained 85% of the measurements. This could possibly be fixed by increasing the window size until there is enough information for an accurate estimation. The algorithms responded very regularly to the widening gap test. When there was a complete time gap of information, the

algorithms were unaffected; however, all of the algorithms diverged when one sensor was active during the information gap. The break down occurred around a gap size of ten seconds. This break down would be easy to fix by ignoring time windows with only one sensor or just increasing the algorithm window size. Lastly, the error model tests revealed that under realistic conditions, the BDNLO will succeed with large bias and time drift values but is also limited to a growing drift instability. Many problems with large bias-drift values have shown to be potentially mitigated by altering algorithm exit conditions and implementing a random start when divergence has been detected as seen in Appendix B.

In each of the algorithms the resulting NEES has suggested that the actual localization error is not actually Gaussian but similar and close enough to use 95% confidence intervals. There are a couple of things that could be producing this type of error. One is the binary nature of the NLO algorithm. If the algorithm succeeds and converges the NEES values tend to fall within an expected range of a chi-squared distribution. However, if the algorithm does not converge well, or the sensor bias is bad enough to create biased estimates, the NEES is much larger than it should be and creates a longer right tail than a chi-squared distribution. After taking the large divergent values out of the statistical analysis as seen in the histograms in Appendix C, there is still a different simulated NEES distribution than a chi-squared. This is likely to be a result of the higher order path elements and bias drift parameters being estimated by the algorithm not being fully accounted for in the confidence estimate. This may explain the different shaped distributions seen for each algorithm in Appendix C where there tends to be a more rounded left tail and a large spike close to the distribution centroid. Regardless of what is causing the difference in distribution, 95% confidence ellipse calculated from the covariance intersection are still accurate. Expanding or shrinking the confidence ellipses could cause problems at very high and low confidences. The binary nature of the NLO algorithm could be fixed by the Random Start addition discussed in Appendix B which

would highly simplify the statistical analysis.

5.3 Algorithm Comparison

Much of this thesis focused on the KMAVNLOCI, BDMOD and BDNLO algorithm. The KMAVNLOCI was developed and tested as well by Sprang as well as the Static NLO, VNLO, VNLOCI, and the KMVNLOCI. The two objectives of this thesis and Sprang are to create the most accurate trajectory estimates and to create the most accurate and honest representation of confidence. Sprang tended to focus more on creating better confidence estimates than accuracy while this thesis did not change the confidence metric much at all. Under the NEES metric, it has been determined that the VNLO has on average produced the best confidence estimates at an average close to the ideal NEES value of three. As the algorithm included the full kinematic model to increase accuracy, the confidence estimates became under confident. The estimates also became more accurate while the algorithm models became more complex.

The new BDMOD and BDNLO algorithms reflect the steady decline in error shown in the development of algorithms discussed in Sprang. The KMAVNLOCI in Sprang was the most accurate algorithm close to the static NLO for simple paths and far above all the algorithms in the simulated complicated paths. Given convergence, the BDMOD and the BDNLO have been shown to produce more accurate estimates than the algorithms developed in Sprang. This is at the large assumption that the sensors being used can be reflected with the Bias-Drift model discussed and used in this thesis. As shown by the test in section 4.4, if the magnitudes of the sensor bias and drift are small enough, the Bias-Drift additions to the NLO algorithms will on average be insignificant with a much higher performance variance. The BDNLO will even create worse estimates on average for higher bias instability values even for an existing bias-drift error model. Given realistic

sensors, the BDNLO and the BDMOD should on average help the NLO algorithms developed by Sprang unless the Bias-Drift IMU error is not valid or better corrected by some other means.

As the KMAVNLOCI created more under confident estimates in comparison to the VNLO, the bias-drift modifications created even more under confident estimates. The confidence estimates may change slightly but the estimate accuracy produced by the new NLO algorithms is improved much more than the confidence accuracy. Creating a more accurate estimate while using the old confidence method will only create more under confident estimates. As shown by the discussion and the histograms in Appendix C, more accurate estimates will shift the chi-squared like NEES distribution left and in a more prominent spike when convergence is likely or a wider spike when convergence is unlikely. Including the Bias-Drift into the estimation model creates much larger right tails in the distribution as a result of the occasional bad bias-drift estimate that can significantly and unknowingly create over confident estimates. Even if those cases only occur a small amount of the time, it continues to shift the confidence estimation further away from a Gaussian system. These algorithm performance differences only occur at small probabilities and do not affect the 95% confidence estimate but should be considered in the interpretation of results if a higher percentage confidence estimate is desired.

5.4 Conclusion

The BDMOD and BDNLO are an easy ways to correct for a sensor bias without any prior knowledge of that bias and time drift. The BDNLO is a great method when there is enough of a bias and drift to significantly affect the localization estimate, otherwise the performance average will be similar to KMAVNLOCI performance except with much more performance variance. For marginal improvement BDMOD is a great method when

the system created by the measurements is too unstable for the BDNLO algorithm to converge and when the KMAVNLOCI is close enough to create a likely Bias-Drift estimate. The new Bias-Drift algorithms do not significantly affect the confidence accuracy either. For more simple trajectories the BDNLO creates more under confident estimates and over confident estimates for more complicated trajectories. The NLO algorithm is a fantastic way of solving complicated problems that may or may not have a closed solution and can be altered to fit many different problems. As seen by the incremental improvements in the field, the effectiveness of the AOA NLO depends on the accuracy of the error and path model.

5.5 Future Work

There are many problems and untested solutions to improve the performance of the different NLO algorithms presented. The NLO algorithms can fail from not converging, as seen in much of Chapter IV. Divergence can be caused by a dominant sensor as observed in section section 4.4, large time gaps containing measurements from only one sensor, as seen in section 4.4, or very large Bias-Drift values as seen in 4.4 for the BDNLO. Appendix B also discusses a new initialization method that could potentially grantee convergence with enough computational power.

Divergence caused by a dominant sensor or time gaps of measurement with only one active sensor could be solved by creating a smarter windowing function. Ideally, measurement windows should only include measurements that are close to the estimated point in time and include an adequate mixture of sensor measurements from enough sensors to create a confident estimate. Many measurements from only one sensor are relatively useless unless there is another sensor perspective to draw depth from. That depth information is lost as an estimation window loses information from all but one

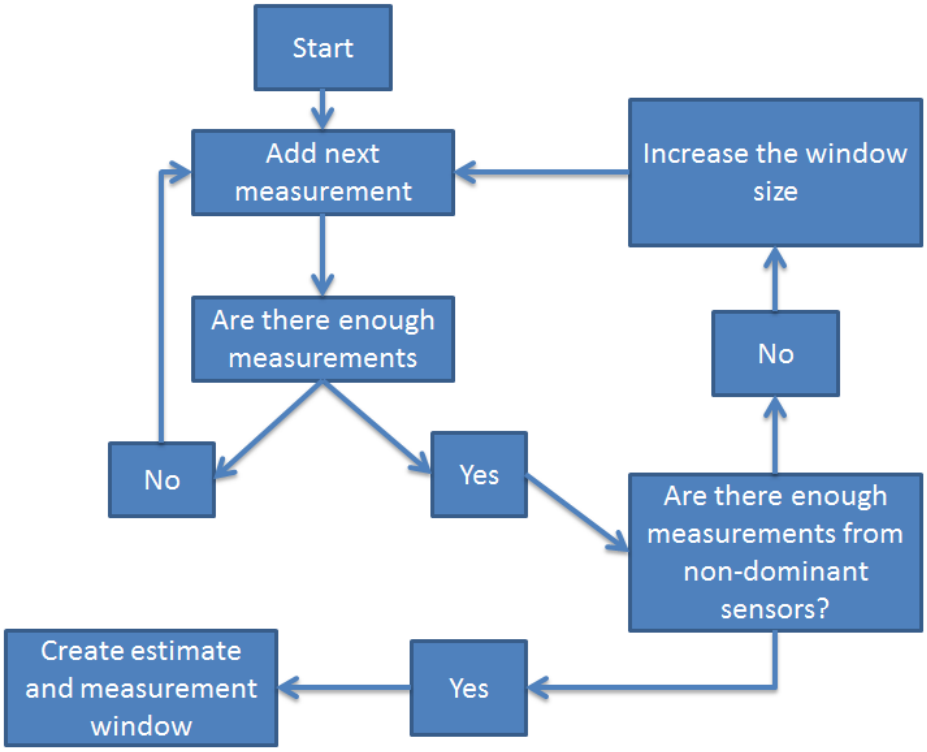

Figure 41. Smart Windowing Flow Chart.

perspective. The current windowing scheme creates estimates spaced in time when there are enough total measurements for that point and a minimum number of sensors involved in the measurement. A new scheme could be developed to choose estimation points spaced according to sensor visibility and then expand windows in time to incorporate enough measurements with a maximum time size for that window. This scheme could fix many of the problem scenarios discussed in this thesis. A conceptual flow chart describing how this might be accomplished can be seen in Figure 41.

The NLO only works if there is some convergence. It's easy to see when the NLO diverges in the simulations that have been run, not only will the residual rise, but the object trajectory will be estimated on the edge of the solar system. This creates a sort of binary performance metric which was corrected in the analysis in this thesis. There are many things that may cause the NLO algorithms to diverge. The initial guess could be

inaccurate enough that the algorithm converges to a local minimum or even diverges because the calculated $\Delta\mathbf{X}$ largely oversteps a small and precise global minimum value. The NLO, in essence, attempts to iterate closer and closer to that global minimum. Convergence becomes harder to achieve as object trajectories become further from the kinematic model and sensor error functions become more significant. There are two potential ways of preventing convergence problems for the Gauss-Newton Method used by the NLO. The object can be initialized in another location. The convergence of iterative methods depends on the accuracy of the initial guess. This has had some initial testing as seen in Appendix B with the random start method. The random start method shows potential, but validating the method is computationally limited. The random start method, even restricted to a type of object, could theoretically run forever without attempting an initial guess close enough to the truth to converge. The random start method does show promise. It has been able to converge to solutions for simulations that could not be solved with a triangulation initial guess. The second divergence prevention technique is using smaller step sizes. If the $\Delta\mathbf{X}$ is calculated close to the correct direction, but a very large magnitude due to an unstable system, the NLO will overshoot the global minimum and potentially diverge. At the cost of speed and additional iterations, the $\Delta\mathbf{X}$ could be reduced by a set percentage to avoid over stepping the global minimum. Using a smaller step size will require many more iterations determined by the $\Delta\mathbf{X}$ reduction factor. This method has not yet been tested on the AOA NLO method developed in this thesis.

Appendix A. Line Segmentation

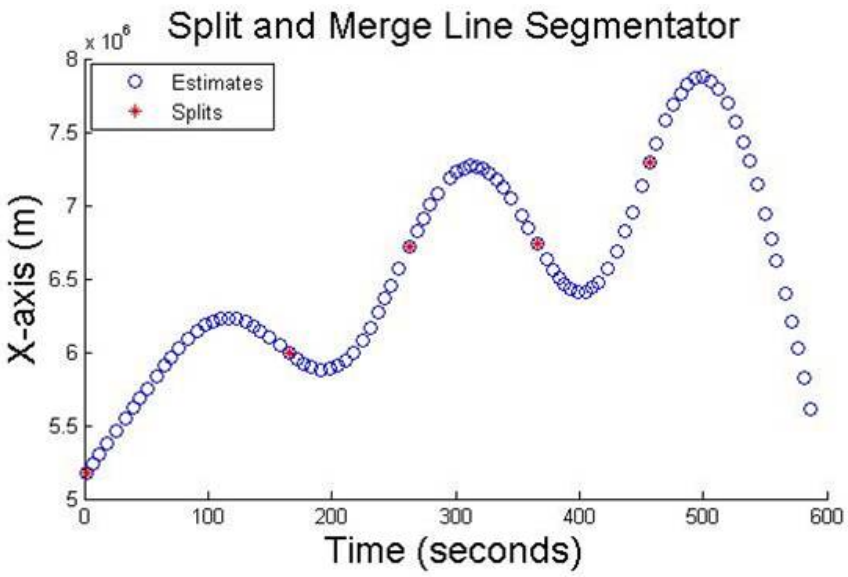

Figure 42. 1D split and merge example.

The Split and merge algorithm described in Chapter 3 functioned as expected on complicated simulated paths. Figure 42 shows a one dimensional cross section of a path (in blue) and the observed splits (in red). The split localization algorithm performed regularly given correct parameters for the given noisiness and measurement density of the path. In most cases, the ideal splits created segments of one kinematic model paths out of a complex path, however; given a simple path that should not be split up, the Split and Merge algorithm tended to have a very high false alarm rate. In many cases where there shouldn't be any splits, the split section found the most likely location and then the merge section many times would fail to locate the false splits.

Once the Split and Merge algorithm was combined with the KMAVNLOCI (creating the SMNLO), a random variable test was done on a complex path where the number of sensors, measurements, sensor configuration, and path complexity were varied much like the test shown in Chapter 4 for the BDNLO and BDMOD. As shown in Figure 43, the split and merge algorithm on average did not affect the performance of the NLO

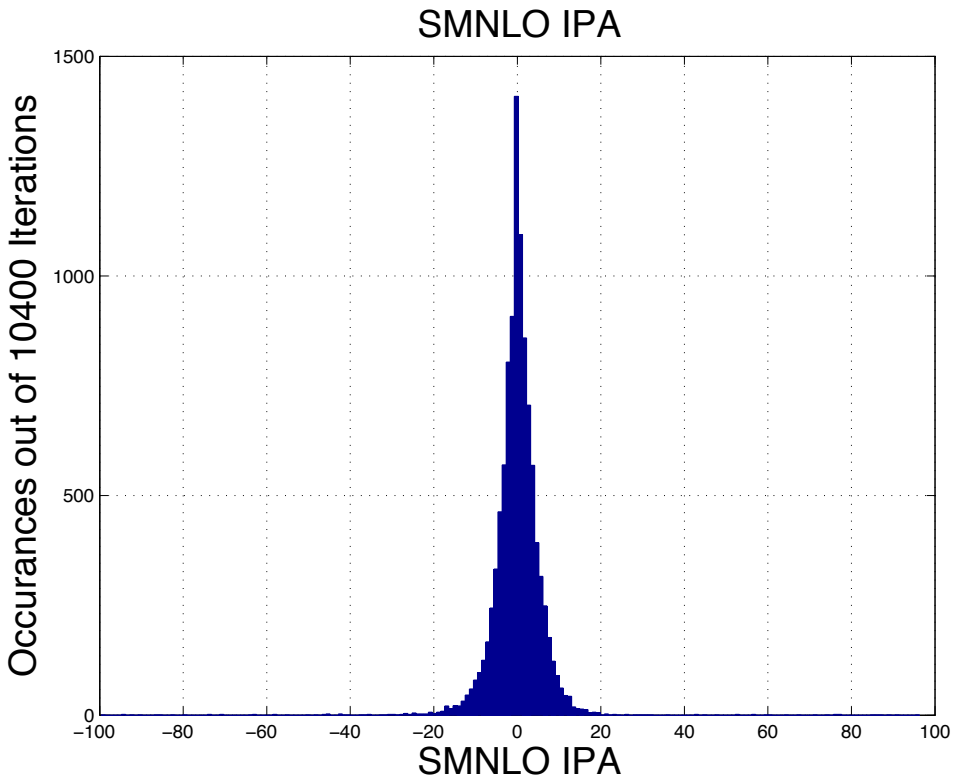

Figure 43. 1D split localization example.

significantly. On average the increase in performance was zero. In most cases there was little to no change in the algorithm. The SMNLO hurt performance equally as often as it would improve performance.

Even when the performance is shown against path complexity as seen in Figure 44, there seems to be little affect from the Split and Merge Algorithm on the error induced by path complexity. The other NLO algorithms tend to have less accurate estimates with more complicated paths as shown in Chapter 4. The Split and Merge algorithm may not be affecting the localization performance because the KMAVNLOCI already estimates for complicated paths enough that there is not much more to be improved upon. The SMNLO may improve performance within a scenario where the path kinematic model changes much more discretely, but in the sensing scenarios tested there significantly does not seem to be any effective performance change.

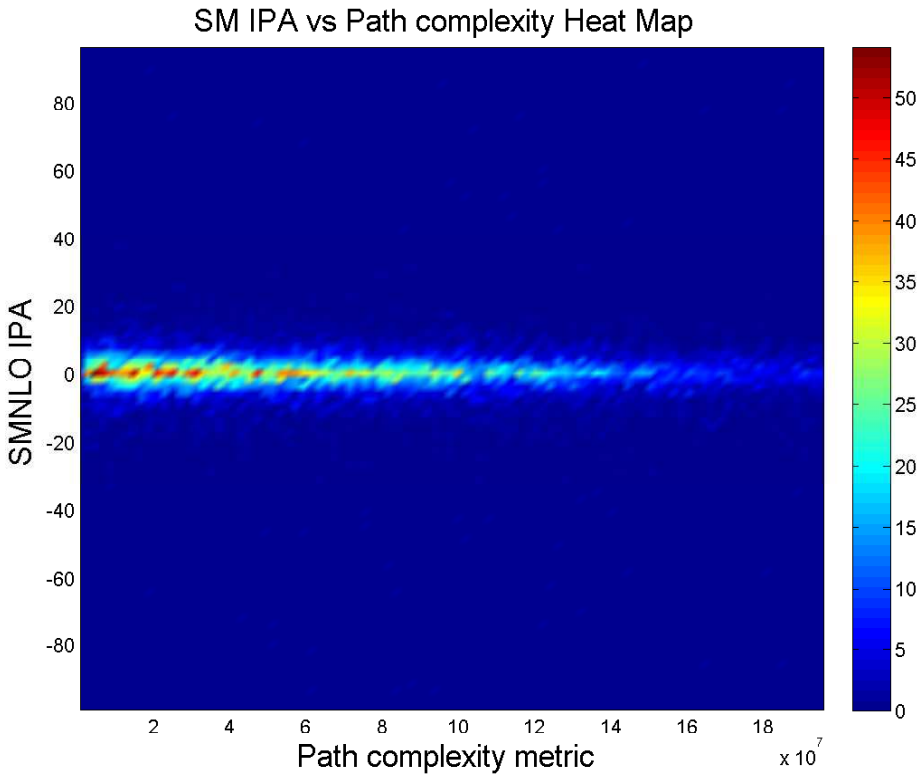

Figure 44. Complexity vs. SMNLO IPA.

Appendix B. Random Start NLO

A new initialization and convergence technique was created to handle larger bias values seen in 4.4 and other divergent cases in the BDNLO. For cases where the BDNLO does not naturally converge, two changes were made: the object trajectory initial guess was made to be random, and the BDNLO residual was allowed to grow a few times before exiting the algorithm. In some cases for the BDNLO the residual would grow and then shrink. This phenomenon occurred during more unstable simulations where the object initial guess was more inaccurate or the measurements were particularly bad. In previous tests, all NLO algorithms were set to exit the NLO iterative loop as soon as residuals began to grow. In some simulations, especially the ones shown in 4.4, the BDNLO became so unstable that it needed to fail and sometimes iterate in good or bad directions before it would converge. In this new scheme, when the BDNLO diverged it would try again with a new randomly generated trajectory. These random trajectories were created by simulating new simple path orbits as they were simulated but with new parameters to narrow down the search space. If the random trajectory was generated close enough to the true path, the BDNLO would not have as much trouble converging. The BDNLO could be attempted on random trajectories until the BDNLO achieves convergence to a small enough residual. Due to the extremely large search space and the trial and error nature of this solution this method could potentially be extremely slow.

This new scheme was tested on the same Bias Factor test discussed in 4.4. In order to complete the simulation, the random start was attempted a maximum of one hundred times per simulation. Below in Figure 45, the scatter plot of the log scale error against the Bias Factor spanning up to a value of ten. The NLO error is displayed with the BDNLO for comparison. With the new method the BDNLO was able to converge and create useful estimates across the range of the Bias factor. Due to a limit of one hundred random start attempts, as the Bias Factor grew large, there are many simulations that just did not

converge. These divergent cases are known to be divergent and it is within reason that given enough random start attempts will eventually converge. The Bias-Drift magnitudes do in fact affect the convergence toughness which is visible in Figure 45, but even more visible when the Drift Factor test was redone with the Random Start addition as seen in Figure 46. The new addition of the Random Start and the altered convergence technique noticeably increased the range of drift magnitudes from 10^{-4} to almost 10^{-2}. At that point the drift magnitude makes it very tough for the BDNLO to converge even with the random guessing. It may be possible in theory; however, as the algorithm requires a more accurate initial guess for success the computational toughness of eventually guessing an accurate guess grows.

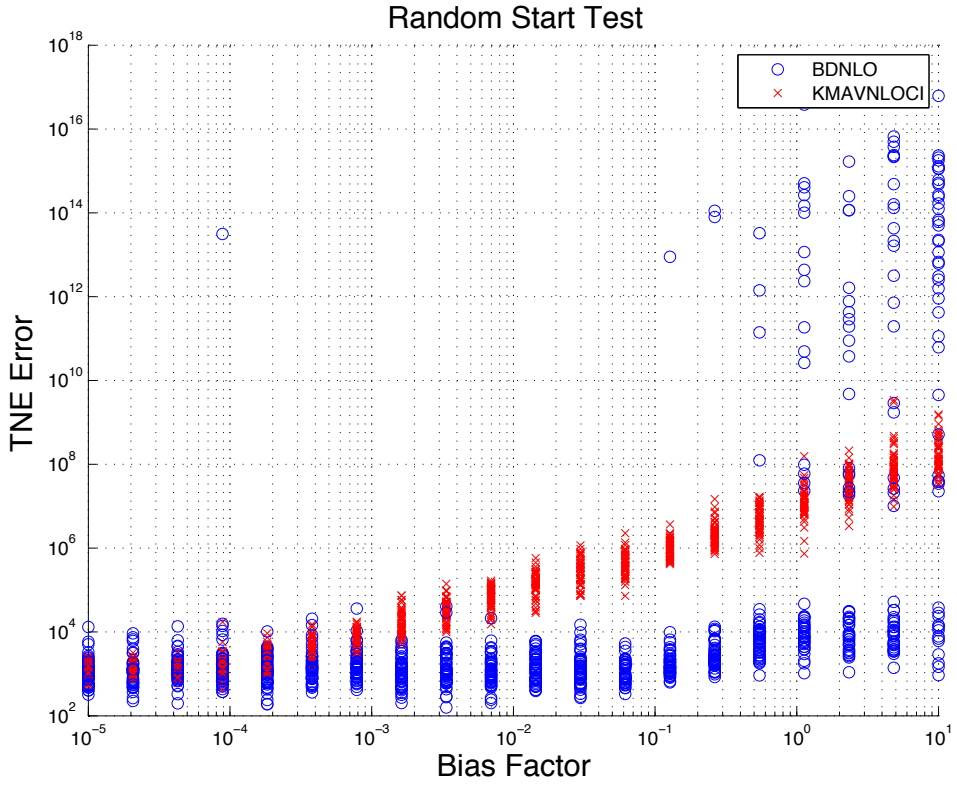

Figure 45. Bias Factor Test with random start BDNLO implemented.

This scheme is computationally taxing and difficult to extensively test as the NLO algorithms have been tested in this thesis. It may be beneficial to further test the addition of the Random Start and the new divergence exit parameters on previous tests to observe if

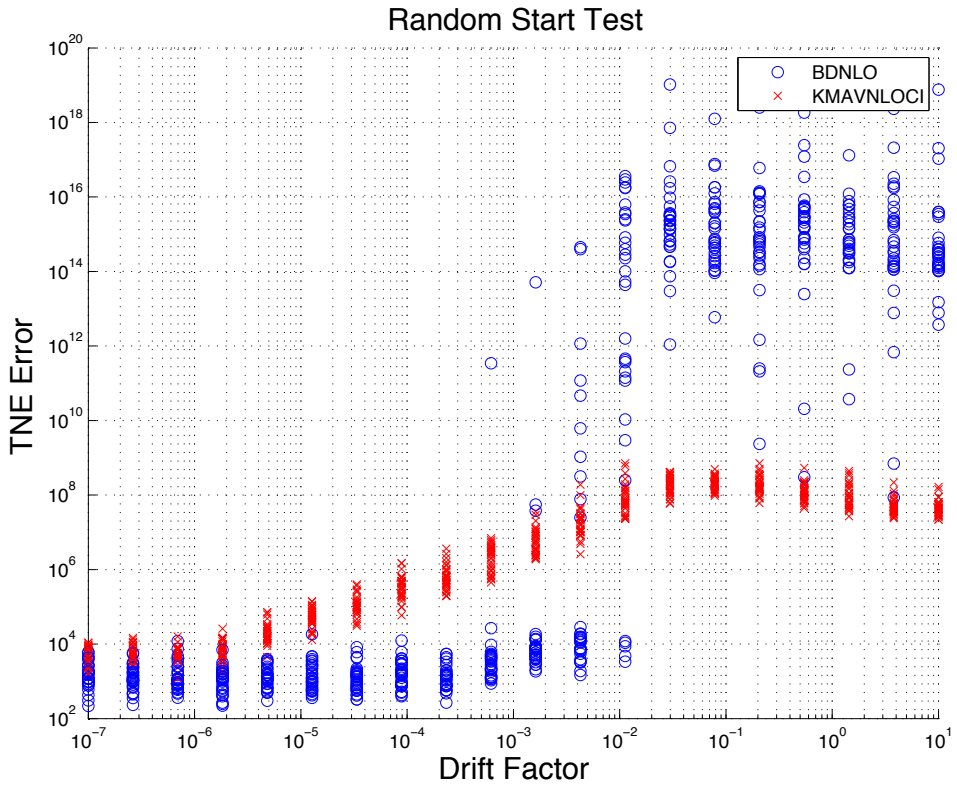

Figure 46. Drift Factor Test with random start BDNLO implemented.

the NLO algorithms can always converge for any situation. Testing this addition would be very beneficial for complicated scenarios where the algorithms have trouble converging. A very large benefit of guaranteeing convergence would be the simplification of the statistical analysis. The localization probability would not be limited to cases given convergence.

Appendix C. NEES Distributions

The NEES values were observed from the variable variation tests. The resulting PDFs estimated from the simulations can be seen in Figures 53-58 along side the assumed chi-squared distribution with three degrees of freedom model. These graphs include the test cases where the NLO algorithms did in fact converge. The known divergent cases would have skewed the distribution as each divergent NEES appeared as an outlier. The resulting estimated PDF does not appear to be chi-squared. Most PDFs contain a more distinguished spike and a much longer right tail. This is most likely the result of the higher order kinematic elements and Bias-Drift being estimated along side the positional estimate. Given a more successful kinematic and Bias-Drift estimation, the location estimates will likely be more accurate than the calculated confidence suggests from the CI. This would explain the sharper spike at lower NEES values than a traditional chi-squared distribution. On the other hand, given a less accurate kinematic model and Bias-Drift estimate, the locational confidence is more likely to be over confident. This would explain the longer and more significant right tail. As well, the BDNLO NEES values are often close to zero rather than the right-shifted spike shown by the KMAVNLOCI and the BDMOD which does not spike until around 2. This is more than likely the result of not handling the unknown bias in the system. As soon as the bias is taken out, it is possible to have estimates close to zero error. Unfortunately, when the BDNLO is less stable and less likely to estimate an accurate bias-drift estimate, as seen in the complicated path case in Figure 58, a larger right tail forms for the inaccurate cases.

There are two implications created by this finding. One, on average the error in the higher order kinematic estimations is not actually Gaussian. If it was in fact a Gaussian probability, the NEES could be accurately described as a chi-squared distribution. Second, the error ellipsoids produced by the CI must be interpreted in a fundamentally different way. They cannot be accurately described as a set multivariate Gaussian distribution. On

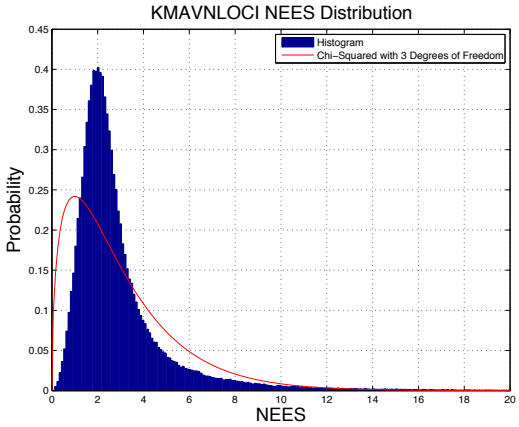

Figure 47. Distribution of the NEES from a KMAVNLOCI Simple Path

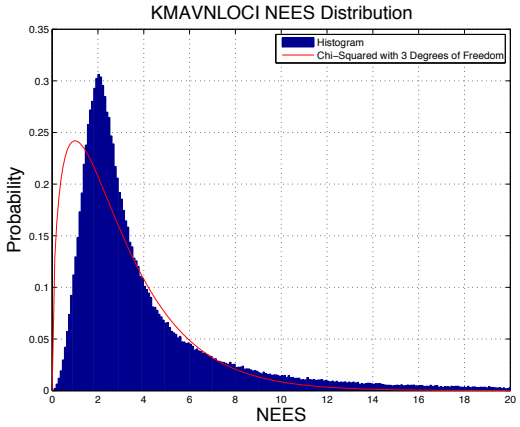

Figure 48. Distribution of the NEES from a KMAVNLOCI Complex Path

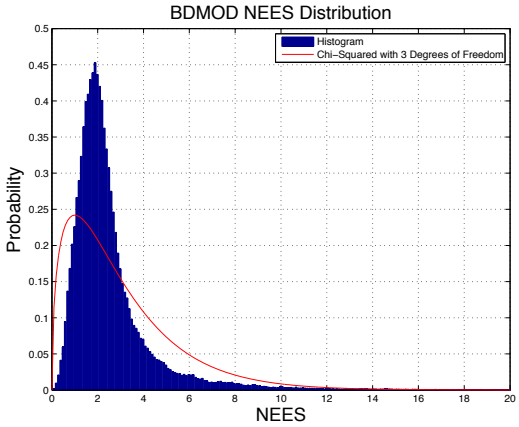

Figure 49. Distribution of the NEES from a BDMOD Simple Path

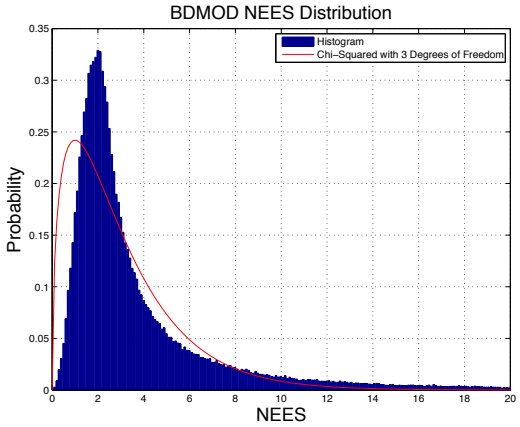

Figure 50. Distribution of the NEES from a BDMOD Complex Path

the plus side, treating the error ellipsoids as a 95% confidence region can sometimes be justified. As stated in Chapter 4, the CDF still reflects around the same 95% confidence region in the simple case. If the error ellipse were to shrink or expand it may run into a statistical inaccuracy, especially if the error ellipse is expanded to more significant confidence regions close to 100%.

Some of the problems existing in the NEES metric are fixed by adding acceleration and velocity components in the AVNEES. The resulting histograms from the new confidence estimate can be seen in Figures 53-58. Much of the right tails in the complicated path tests have been significantly diminished to a point where it is no longer a

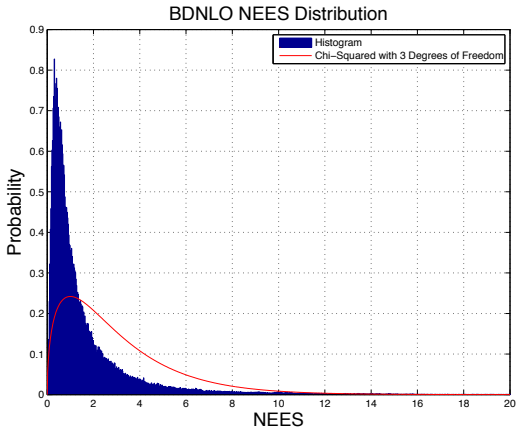

Figure 51. Distribution of the NEES from a BDNLO Simple Path

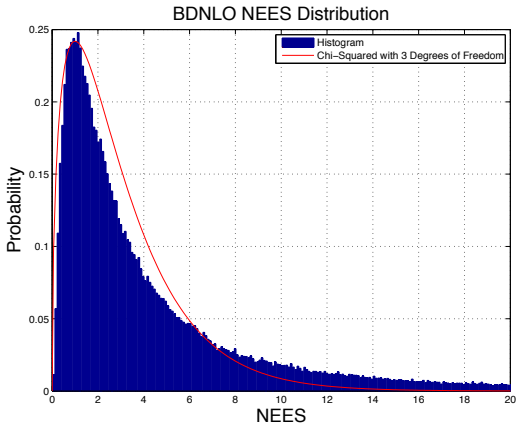

Figure 52. Distribution of the NEES from a BDNLO Complex Path

large concern. Especially in the BDNLO case where the the NEES right tail is significant. Using the AVNEES graphically appears be closer to a chi-squared distribution than the NEES. This suggests that the varying time squared element used in creating the AVNEES may have been affecting the NLO error distributions. The AVNEES appears to have taken out some of those squared element in the error distribution, but there is still some more squared elements remaining either from an inaccurate covariance estimate or unobserved elements determined by the NLO convergence performance.

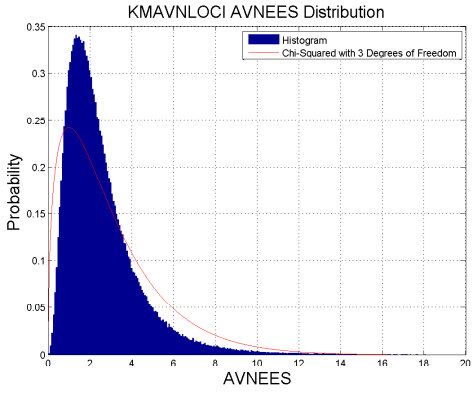

Figure 53. Distribution of the AVNEES from a KMAVNLOCI Simple Path

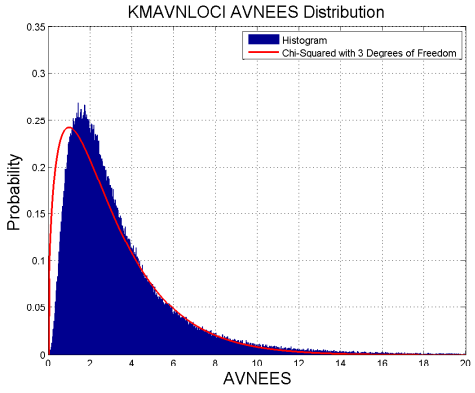

Figure 54. Distribution of the AVNEES from a KMAVNLOCI Complex Path

A good metric to determine which simulated distribution is more similar to the assumed model is the Kullback-Leibler Divergence (KLD). The KLD measures the

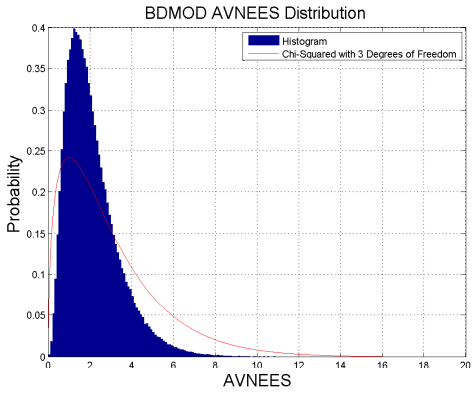

Figure 55. Distribution of the AVNEES from a BDMOD Simple Path

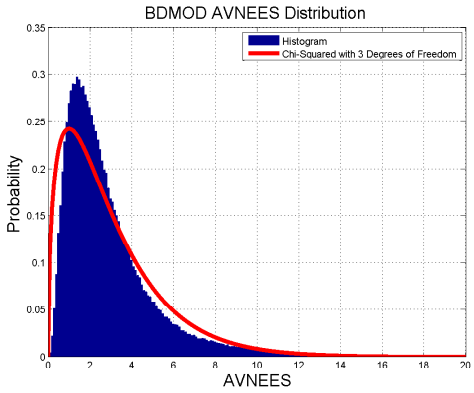

Figure 56. Distribution of the AVNEES from a BDMOD Complex Path

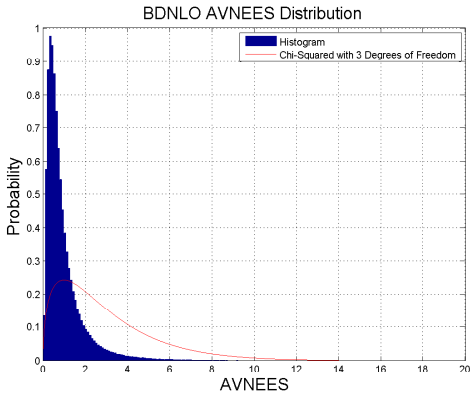

Figure 57. Distribution of the AVNEES from a BDNLO Simple Path

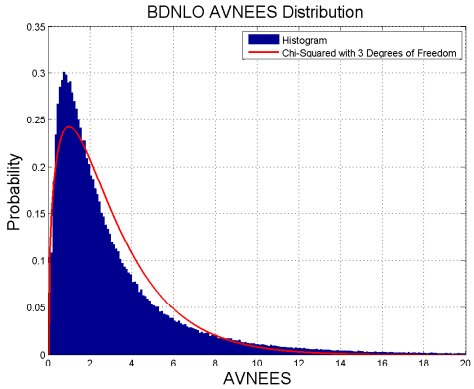

Figure 58. Distribution of the AVNEES from a BDNLO Complex Path

statistical difference between two PDFs and the lower the number the closer the two distributions are to each other by measuring the log difference between the two PDFs as seen in

$$KLD = \int P(x) log \left(\frac{P(x)}{Q(x)} \right) dx \tag{28}$$

Where $P(x)$ is the ideal model PDF and $Q(x)$ is the actual PDF. The KLD specifically measures the relative Shannon Entropy between the assumed model and the measured truth. This metric then becomes an ideal way to adequately determine how

much information is lost by assuming the estimation distribution models proposed for these algorithms. The larger the KLD the more information is lost by the model [20] [21]. The KLD for the different simulated PDFs can be seen in table 3. The change to the AVNEES improves the KLD in every case except for the BDNLO in the simple path test. The AVNEES as well drastically improves the bias-drift algorithms for the complex path tests.

Table 3. Kullback-Leibler Divergence.

KLD	Simple Path	
	NEES	AVNEES
KMAVNLOCI	3.02	1.84
BDNLO	3.98	8.40
BDMod	3.16	2.85
KLD	Complicated Path	
	NEES	AVNEES
KMAVNLOCI	3.77	3.19
BDNLO	20.70	0.97
BDMod	2.62	0.59

Appendix D. Unbalanced Sensor Sampling Test

This section of the appendix contains more results from the unbalanced sensor measurement sweep tests. This was done on total 3-6 sensors for the BDNLO, BDMOD and KMAVNLOCI algorithms. While each test regularly begins to fail and diverge when the dominant sensor contains 85% of the measurements, it appears that the more sensors used, the lower the chance of divergence will be before that unavoidable failure point. It is important to note that even when the dominant sensor contains below 80% of the measurements there are still some large jumps in error. This shows that there is always some probability of divergence in this algorithm, and that probability grows as the sensor measurement become more unbalanced. If this test was attempted again where the results were averaged over more than 5 paths, there would probably be a noticeably growing variance in the performance building up to the divergent 85% area regularly shown in the figures below.

Some future work that could be done in this area could include observing the affects of down sampling a dominant sensor. There could a point where down sampling could be beneficial to the algorithm's performance. More indication that down sampling could fix this problem stems from earlier tests which suggested that containing a large amount of measurements is not a significant factor on potential performance if window sizes are chosen in the same way. This needs to be tested before the down sampling solution is confirmed. Another explanation of divergence could be caused by the dominance of one sensor on the mathematical process, there may not be enough depth information from the other sensors to make accurate estimates and that is why the algorithm is diverging and not necessarily from a dominant bias.

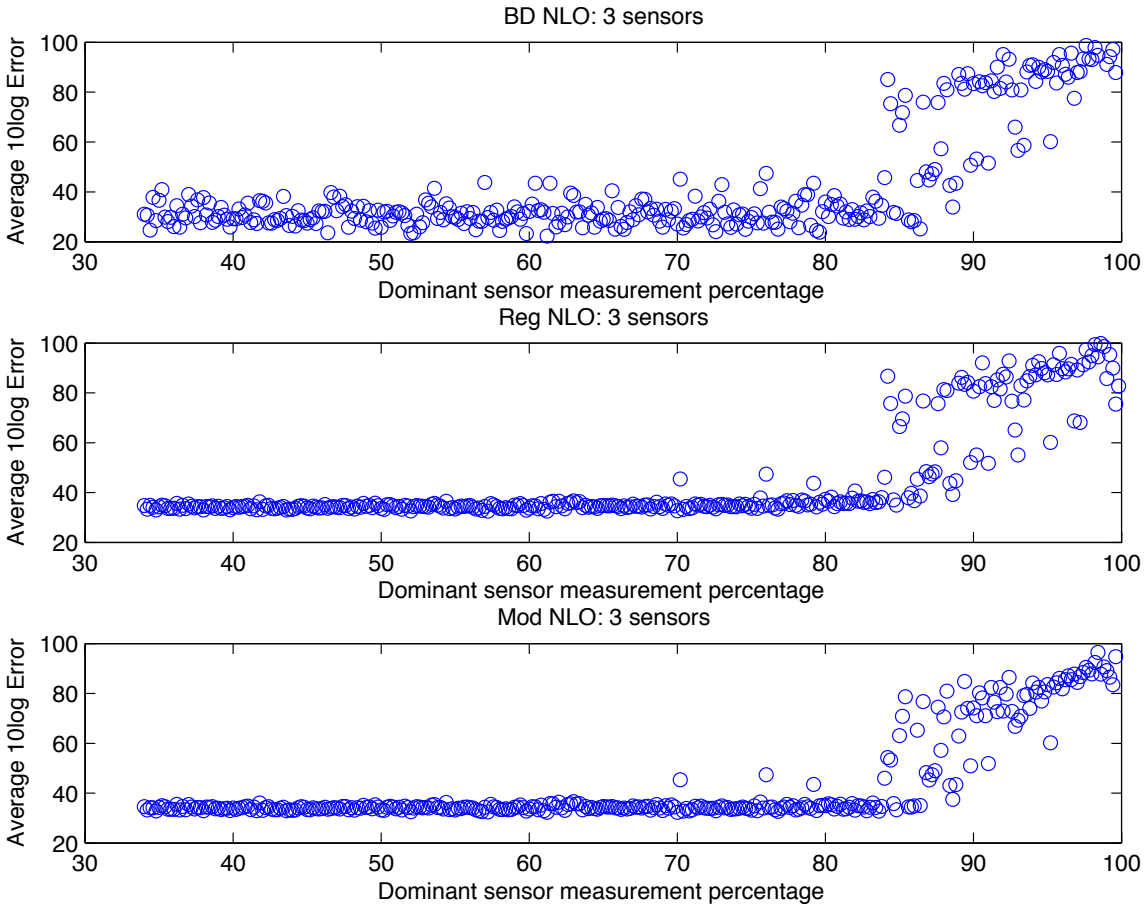

Figure 59. Unbalanced sensor measurement concentration sweep: 3 Sensors, Simple path.

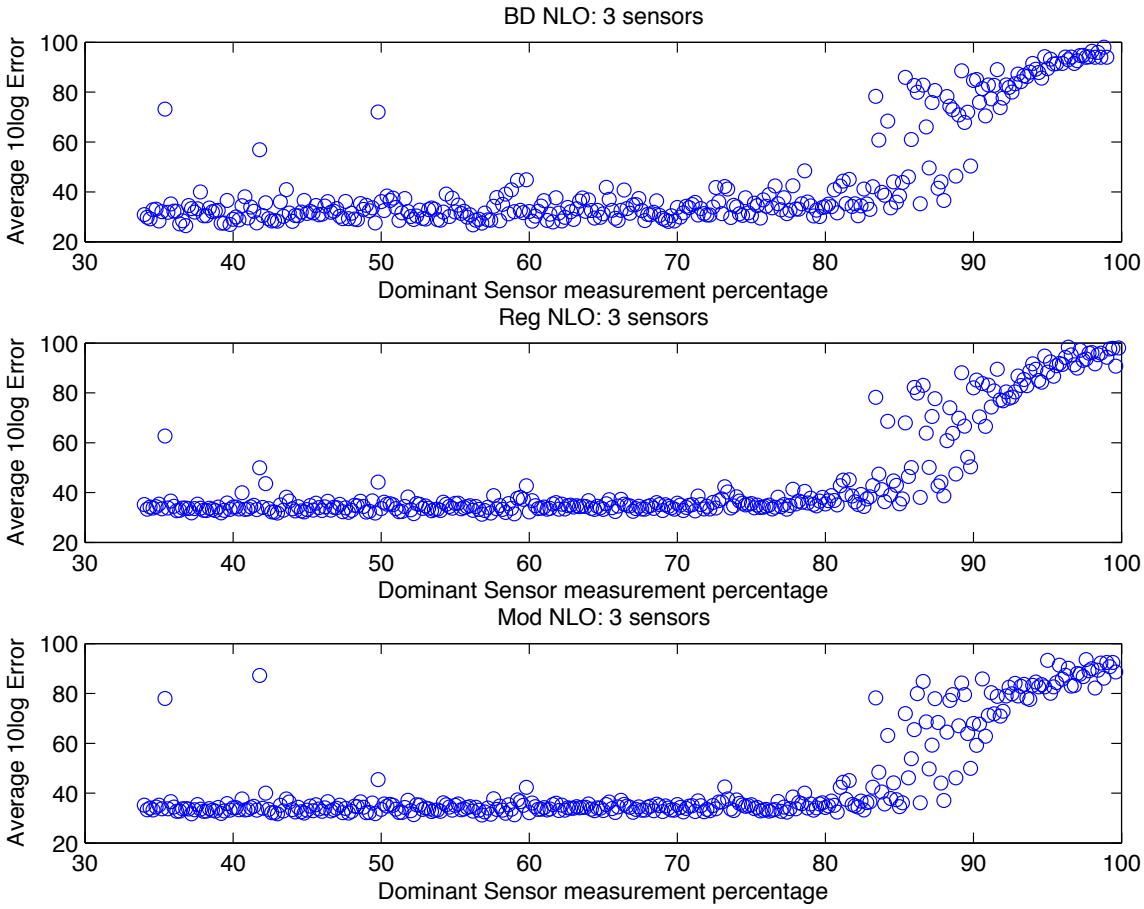

Figure 60. Unbalanced sensor measurement concentration sweep: 3 Sensors, Complicated path.

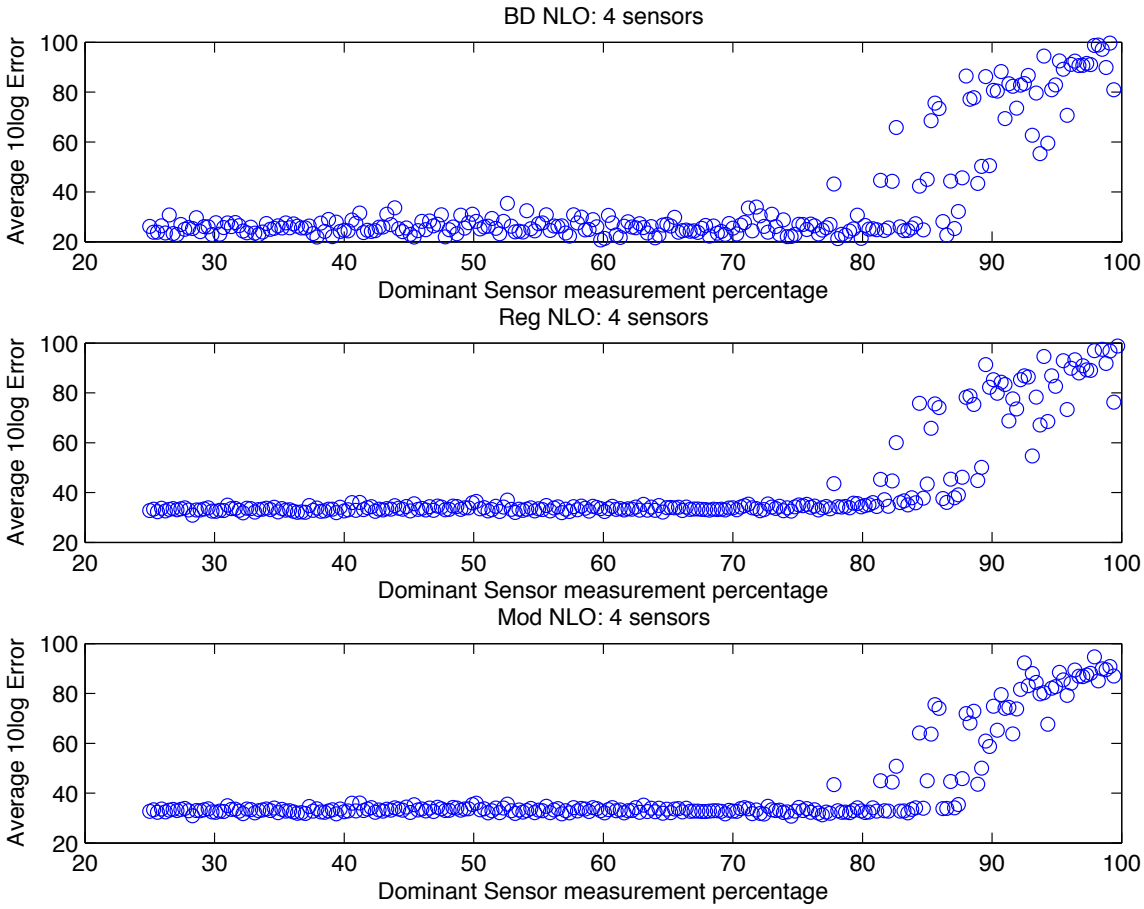

Figure 61. Unbalanced sensor measurement concentration sweep: 4 Sensors, Simple path.

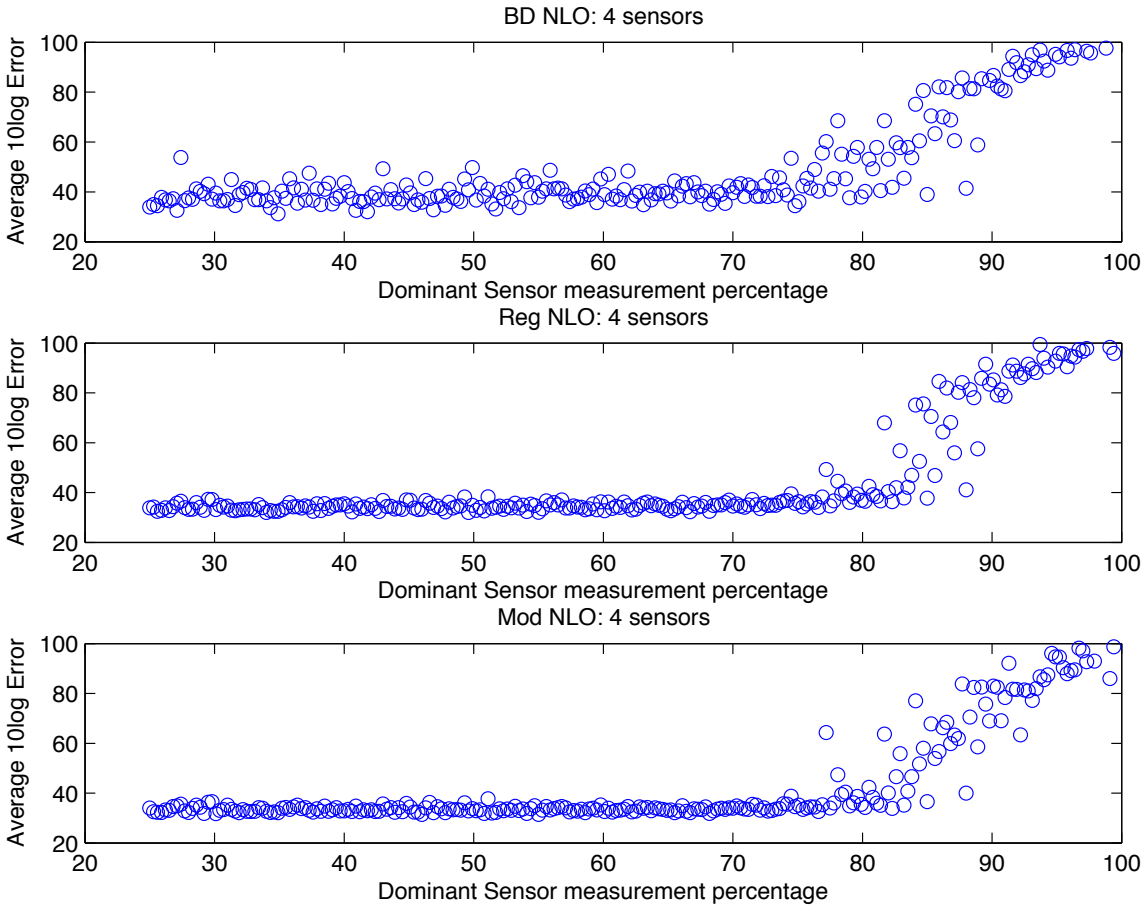

Figure 62. Unbalanced sensor measurement concentration sweep: 4 Sensors, Complicated path.

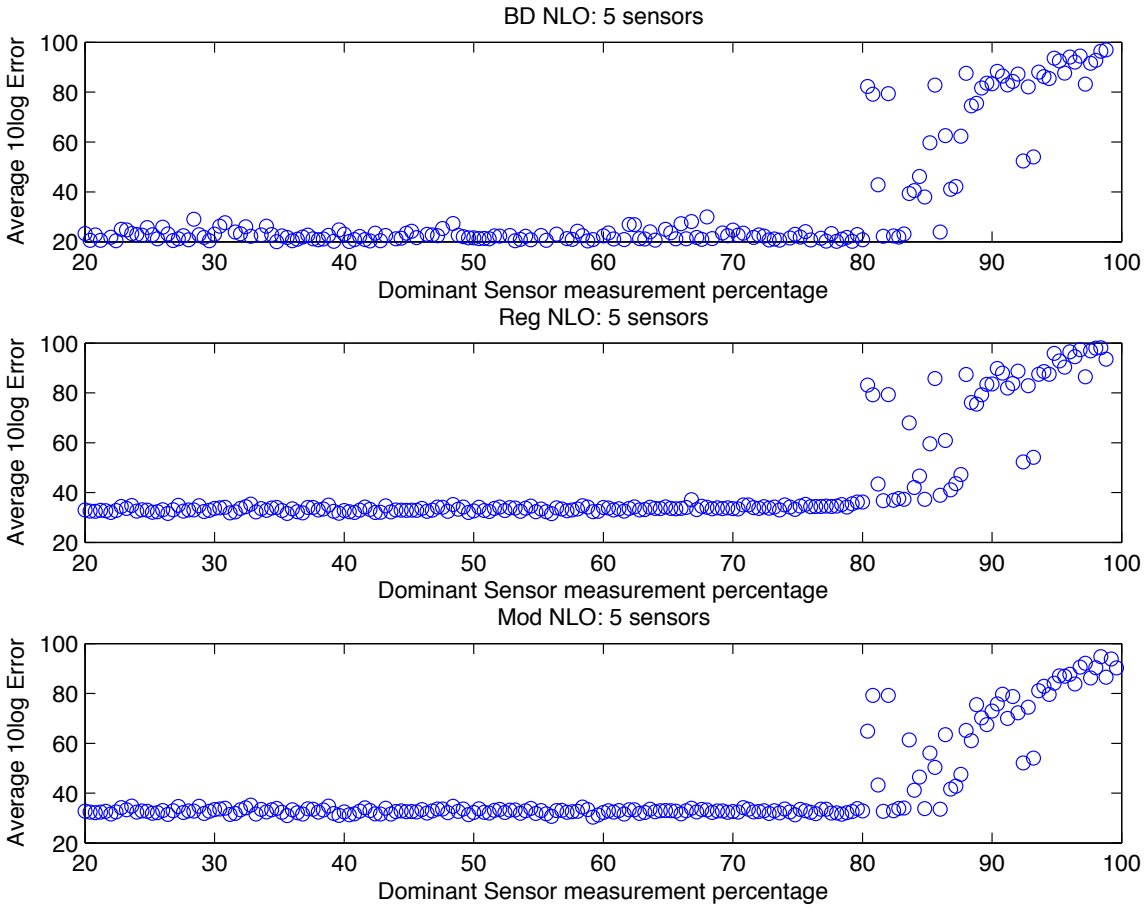

Figure 63. Unbalanced sensor measurement concentration sweep: 5 Sensors, Simple path.

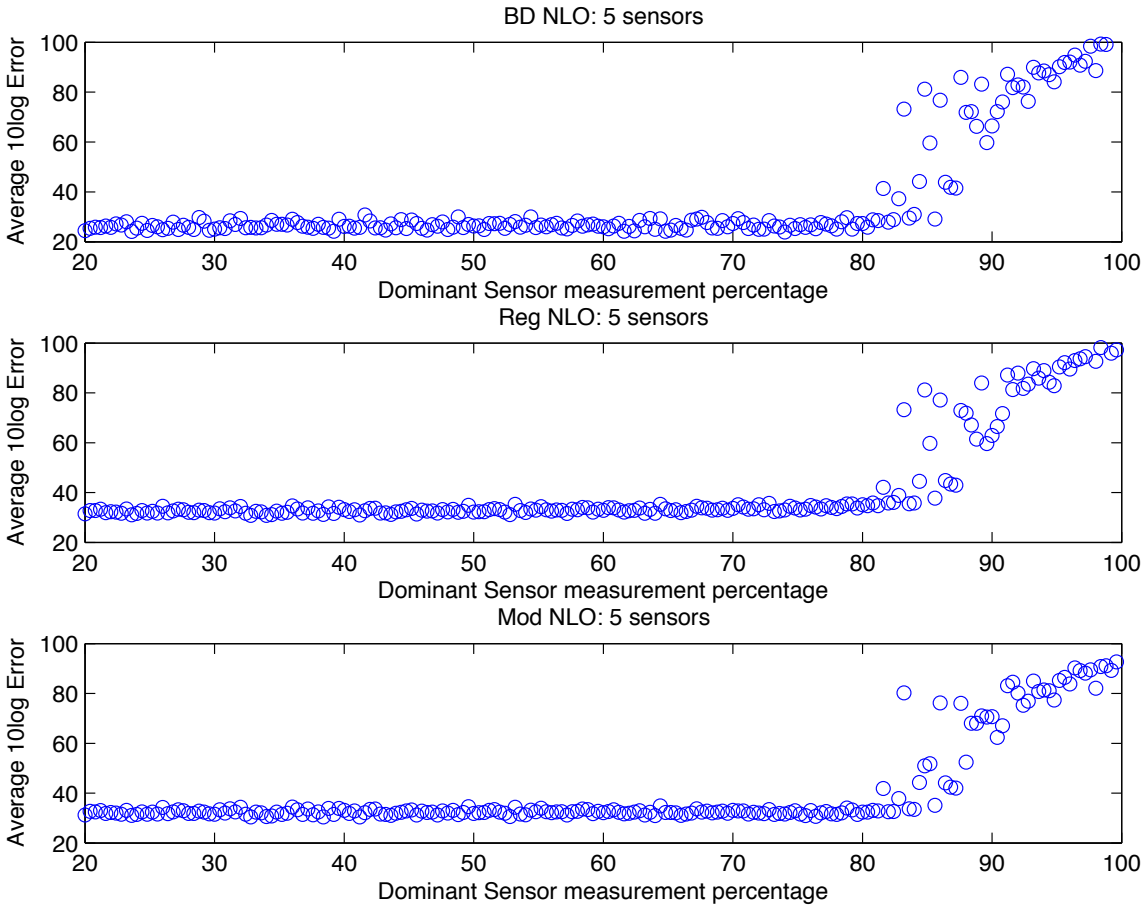

Figure 64. Unbalanced sensor measurement concentration sweep: 5 Sensors, Complicated path.

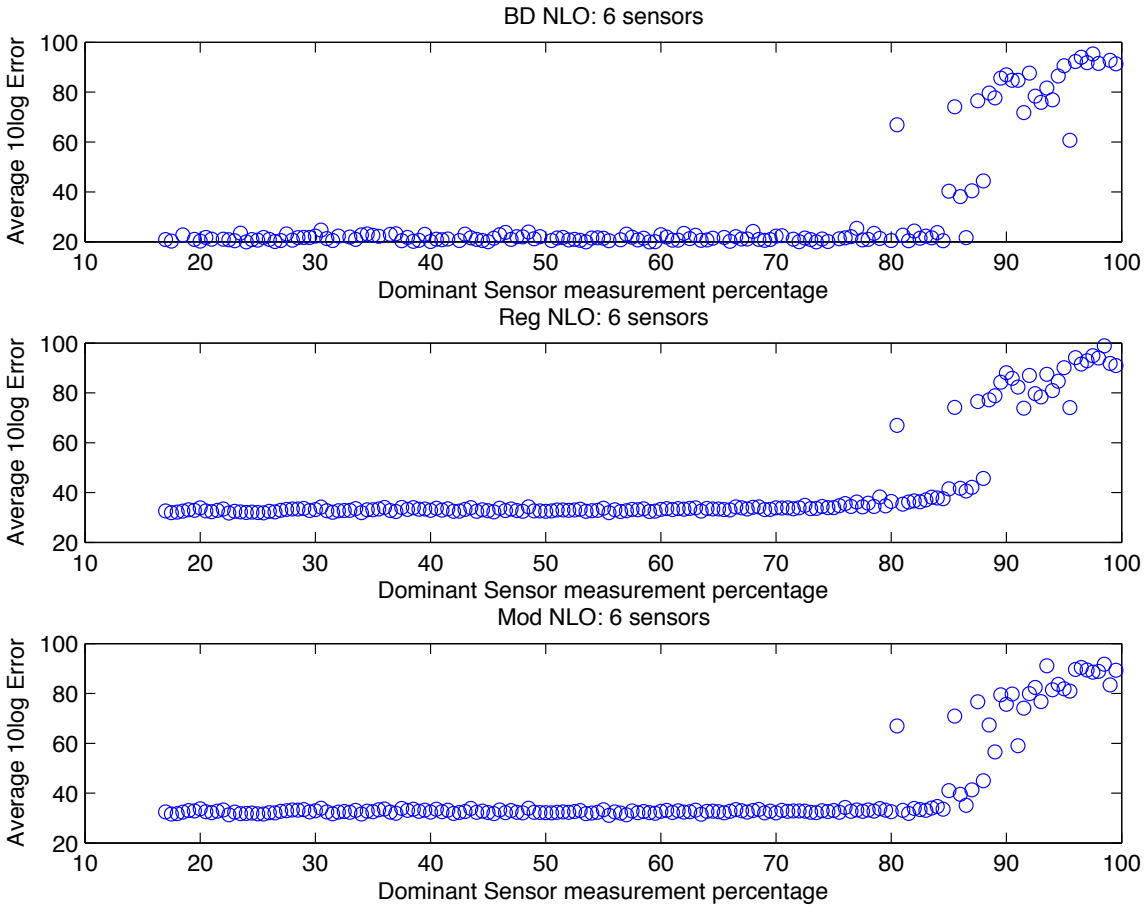

Figure 65. Unbalanced sensor measurement concentration sweep: 6 Sensors, Simple path.

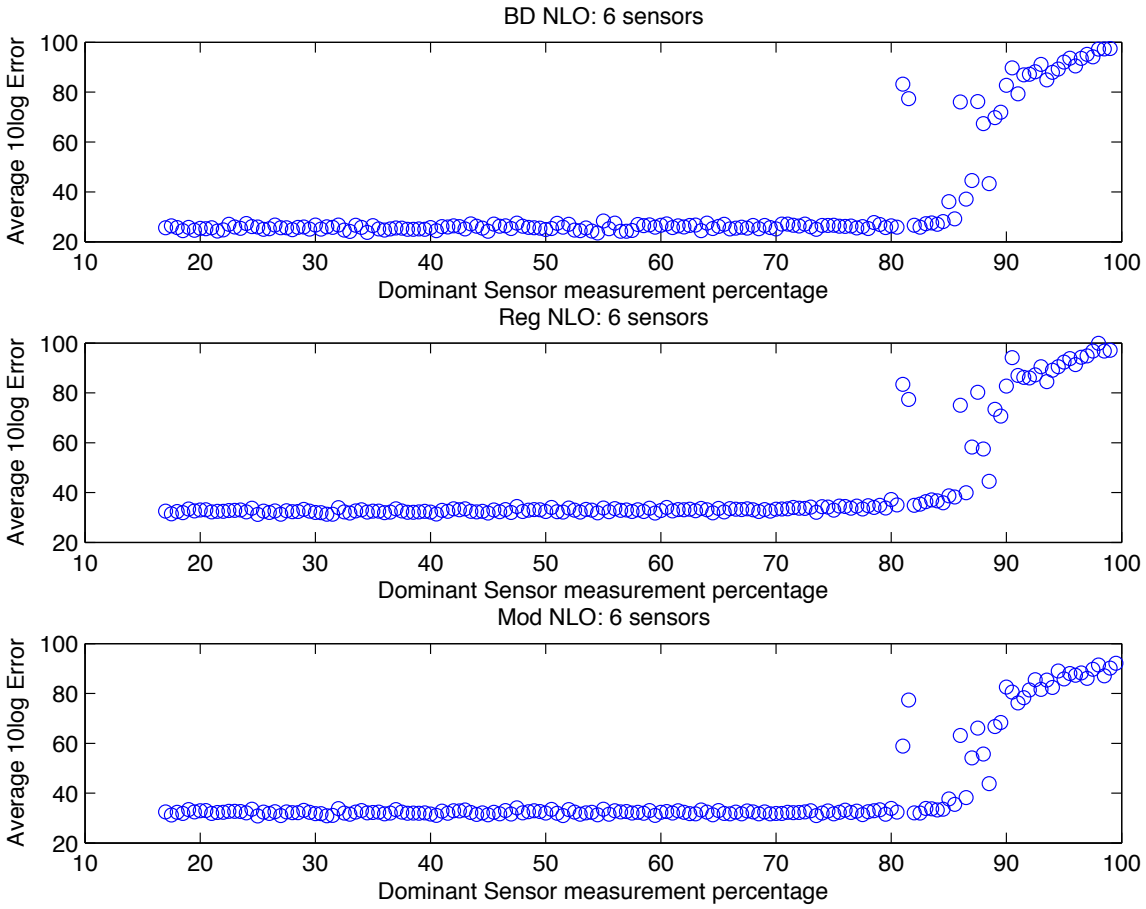

Figure 66. Unbalanced sensor measurement concentration sweep: 6 Sensors, Complicated path.

Appendix E. Visibility Tests

The below figures show the results from the visibility test on a complicated path. The affect appears to be the same on a complicated path as it does for a simple path except slightly less accurate. The BDNLO appears to be slightly more affected by average visibility for complicated paths but is by and large very similar.

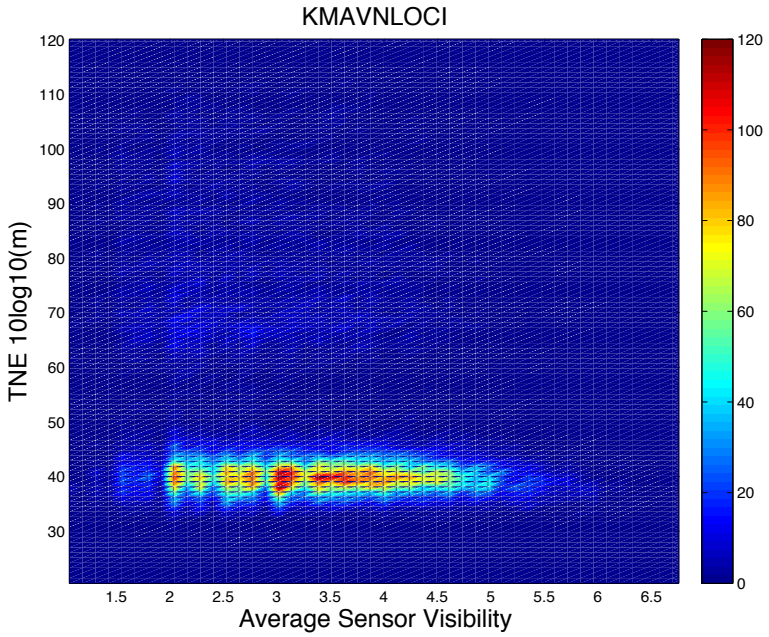

Figure 67. Complex Path random Visibility Test: KMAVNLOCI.

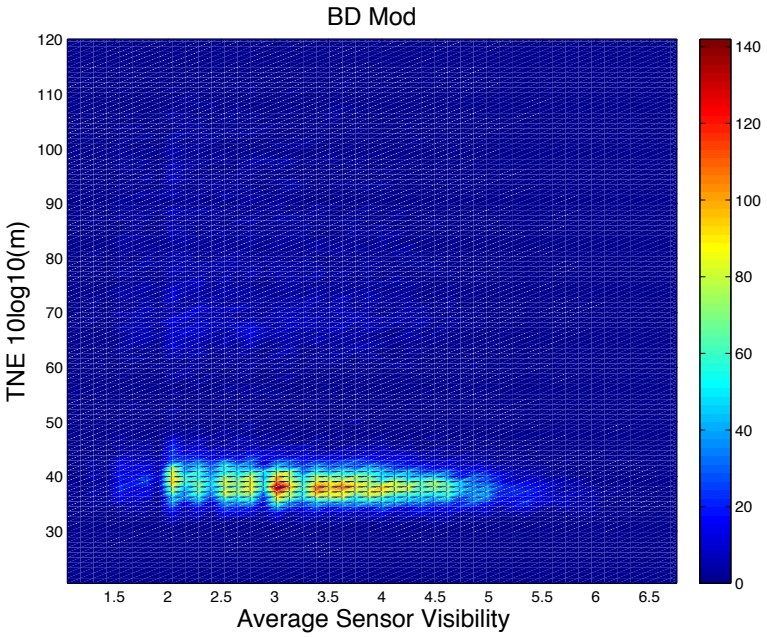

Figure 68. Complex Path random Visibility Test: BDMOD.

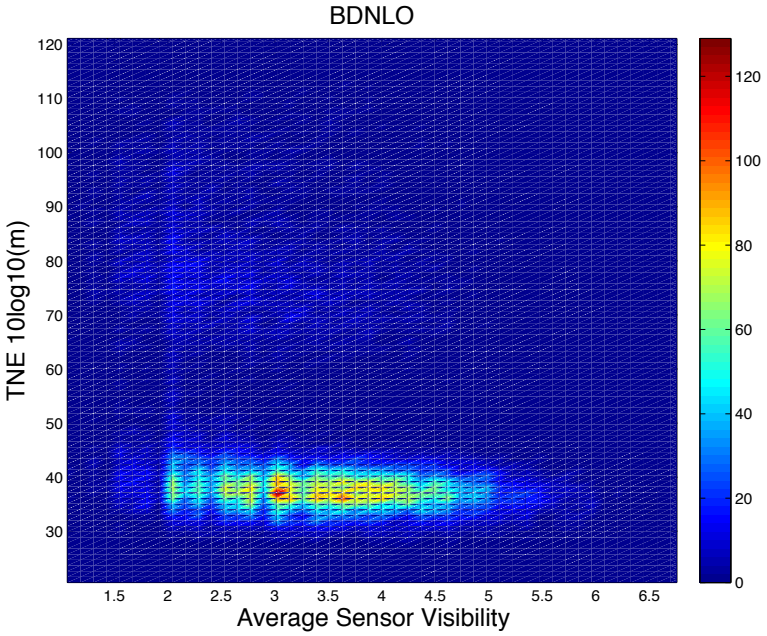

Figure 69. Complex Path random Visibility Test: BDNLO.

Bibliography

[1] J. S. Sprang. “Non-Linear Optimization Applied to Angle-of-Arrival Satellite-Based Geolocation with Correlated Measurements”. MA thesis. Air Force Institute of Technology, 2015.

[2] S. Hartzell. “Non-linear Optimization Applied to Angle-of-Arrival Satellite-Based Geolocation”. MA thesis. Air Force Institute of Technology, 2014.

[3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004.

[4] S.J. Julier and J.K. Uhlmann. “A non-divergent estimation algorithm in the presence of unknown correlations”. In: *American Control Conference.* Vol. 4. 1997, 2369–2373 vol.4. DOI: 10.1109/ACC.1997.609105.

[5] S.J. Julier and J.K. Uhlmann. “Simultaneous localisation and map building using split covariance intersection”. In: *Intelligent Robots and Systems. Proceedings. IEEE/RSJ International Conference on*. Vol. 3. 2001, 1257–1262 vol.3. DOI: 10.1109/IROS.2001.977155.

[6] T. Svantesson and J.W. Wallace. “Tests for assessing multivariate normality and the covariance structure of MIMO data”. In: *Acoustics, Speech, and Signal Processing. Proceedings. (ICASSP ’03). IEEE International Conference on*. Vol. 4. 2003, IV–656–9 vol.4. DOI: 10.1109/ICASSP.2003.1202728.

[7] Oliver J. Woodman. *An introduction to inertial navigation*. Tech. rep. University of Cambridge, Computer Laboratory, 2007.

[8] D. Gebre-Egziabher, J. D. Powell, and P. K. Enge. “Design and Performance Analysis of a Low-Cost Aided Dead Reckoning Navigator”. PhD thesis. Stanford University, Department of Aeronautics and Astronautics, 2001.

[9] J. D. Gautier. “GPS/INS Generalized Evaluation Tool (GIGET) for the Design and Testing of Integrated Navigation Systems”. PhD thesis. Stanford University, Department of Aeronautics and Astronautics, 2003.

[10] A. Wu. “SBIRS high payload LOS attitude determination and calibration”. In: *Aerospace Conference, 1998 IEEE*. Vol. 5. 1998, 243–253 vol.5. DOI: 10.1109/AERO.1998.685827.

[11] J.A. Besada Portas, J.G. Herrero, and G.M. Vela. “New approach to online optimal estimation of multisensor biases”. In: *Radar, Sonar and Navigation, IEE Proceedings* - 151.1 (2004), pp. 31–40. ISSN: 1350-2395. DOI: 10.1049/ip-rsn:20040116.

[12] E. D. Kaplan. *Understanding GPS: Principles and Applications, Second Edition*. Artech House, 2005. ISBN: 1580538940.

[13] S. Kay. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory (v. 1)*. Prentice Hall, 1993. ISBN: 0133457117.

[14] B. Yang and J. Scheuing. “Cramer-Rao bound and optimum sensor array for source localization from time differences of arrival”. In: *Acoustics, Speech, and Signal Processing. Proceedings. (ICASSP ’05). IEEE International Conference on*. Vol. 4. 2005, iv/961–iv/964 Vol. 4. DOI: 10.1109/ICASSP.2005.1416170.

[15] Yaakov Shalom. *Estimation with applications to tracking and navigation*. New York: Wiley, 2001. ISBN: 978-0471416555.

[16] Z. Zhao X. R. Li and V. P. Jilkov. “Practical Measures and Test for Credibility of an Estimator”. In: 2001, pp. 481–495.

[17] C. Panagiotakis et al. “Segmentation and Sampling of Moving Object Trajectories Based on Representativeness”. In: *Knowledge and Data Engineering, IEEE*

Transactions on 24.7 (2012), pp. 1328–1343. ISSN: 1041-4347. DOI: 10.1109/TKDE.2011.39.

[18] Jae gil Lee and Jiawei Han. "Trajectory Clustering: A Partition-and-Group Framework". In: *In Special Interest Group on Management of Data*. Vol. 151. 2007, pp. 593–604.

[19] L. Trefethen. *Numerical linear algebra*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1997. ISBN: 978-0-898713-61-9.

[20] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *Ann. Math. Statist.* 22.1 (Mar. 1951), pp. 79–86. DOI: 10.1214/aoms/1177729694. URL: http://dx.doi.org/10.1214/aoms/1177729694.

[21] J.R. Hershey and P.A. Olsen. "Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models". In: *Acoustics, Speech and Signal Processing. IEEE International Conference on*. Vol. 4. 2007, pp. IV–317–IV–320. DOI: 10.1109/ICASSP.2007.366913.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
24–03–2016	Master's Thesis	Sept 2014 — Mar 2016

4. TITLE AND SUBTITLE

Non-Linear Optimization Applied to Angle-of-Arrival Satellite Based Geo-Localization for Biased and Time Drifting Sensors

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Levy, Daniel E., 2nd Lt, 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-MS-16-M-032

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

Multiple sensors are used in a variety of geolocation systems. When an object does not emit a classical RF signal, AOA measurements become more feasible than TDOA or RSS methods. A NLO method for calculating the most likely estimate from AOA measurements has been created in previous work. This thesis, modifies that algorithm to automatically correct AOA measurement errors by estimating the inherent bias and time-drift in the IMU of the AOA sensing platform. Two methods are created to correct the sensor bias. One method corrects the sensor bias in post processing while treating the previous NLO method as a module. The other method directly corrects the sensor bias within the NLO algorithm by incorporating the bias parameters as a state vector in the estimation process. These two methods are analyzed using various Monte-Carlo simulations to check the general performance of the two modifications in comparison to the original NLO algorithm.

15. SUBJECT TERMS

Localization, Angle-of-Arrival, Non-Linear Optimization, Passive Tracking, Bias, Time-Drift

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. A. Terzuoli, AFIT/ENG
U	U	U	U	124	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4717; andrew.terzuoli@afit.edu