

PERFORMANCE EVALUATION CRITERIA AND ANALYSIS OF NAVIGATION SYSTEMS USING INERTIAL MEASUREMENT UNIT TECHNOLOGY

Kyle J. DeMars, et al.

**University of Missouri System
Missouri University of Science and Technology
300 West 12 Street
Rolla, MO 65409-6506**

26 Jun 2014

Final Report

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.



**AIR FORCE RESEARCH LABORATORY
Space Vehicles Directorate
3550 Aberdeen Ave SE
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776**

DTIC COPY
NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RV-PS-TR-2014-0088 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//SIGNED//

MORGAN BALDWIN
Program Manager

//SIGNED//

PAUL HAUSGEN
Technical Advisor, Spacecraft Component Technology Branch

//SIGNED//

BENJAMIN M. COOK, Lt Col, USAF
Deputy Chief, Spacecraft Technology Division
Space Vehicles Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 26-06-2014		2. REPORT TYPE Final Report		3. DATES COVERED (From - To) 05 Jun 2013 – 28 Apr 2014	
4. TITLE AND SUBTITLE Performance Evaluation Criteria and Analysis of Navigation Systems Using Inertial Measurement Unit Technology				5a. CONTRACT NUMBER FA9453-13-1-0289	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 63401F	
				5d. PROJECT NUMBER 5083	
6. AUTHOR(S) Kyle J. DeMars, Jacob E. Darling, Michael A. Waltemate, and Samuel J. Haberberger				5e. TASK NUMBER PPM00016475	
				5f. WORK UNIT NUMBER EF011135	
				8. PERFORMING ORGANIZATION REPORT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES) University of Missouri System Missouri University of Science and Technology 300 West 12 Street Rolla, MO 65409-6506					
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Space Vehicles Directorate 3550 Aberdeen Ave., SE Kirtland AFB, NM 87117-5776				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RVSV	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RV-PS-TR-2014-0088	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT A primary aspect involved in the navigation of a wide variety of vehicles, such as ground vehicles and aircraft, is the use of inertial measurement units. The purpose of the inertial measurement unit is to provide an internal measure of the linear acceleration and angular velocity of the vehicle in order to be used by a navigation system in the dead-reckoning of the vehicle's position and attitude. Due to the nature of sensors, the measurement of the linear acceleration and angular velocity are corrupted by several error sources, such as bias and noise, amongst others. In order to assess the coupled impact of these error sources on the navigation accuracy for an arbitrary vehicle, a simulation and analysis tool is developed that can model several inertial measurement unit mechanizations, simulate the error sources encountered in the acquisition of measurement data, emulate the navigation software, and perform a range of statistical analyses via Monte Carlo simulation. The developed tool is applied to a ground vehicle navigation scenario in order to assess the performance of different sensors and provide conclusions as to when each sensor should be used to achieve the best navigation performance.					
15. SUBJECT TERMS Inertial Measurement Unit (IMU) Modeling, Dead-Reckoning, Navigation, Monte Carlo Simulation, Uncertainty Analysis					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area
			Unlimited	50	Morgan Baldwin

(This page intentionally left blank)

TABLE OF CONTENTS

Section	Page
List of Figures	ii
1.0 SUMMARY	1
2.0 INTRODUCTION	1
3.0 METHODS, ASSUMPTIONS, AND PROCEDURES	3
3.1 SAIMUN Overview	3
3.2 IMU Modeling	5
3.2.1 Startup Bias	6
3.2.2 Bias Instability	6
3.2.3 Velocity Random Walk	7
3.2.4 Scale Factor Error	8
3.2.5 Axes Misalignment Error	9
3.2.6 Axes Nonorthogonality Error	9
3.2.7 Quantization Error	9
3.3 Allan Variance	10
3.4 IMU Mechanizations	13
3.4.1 Strapdown Mechanization	14
3.4.2 Local-Level Mechanization	17
3.4.3 Space-Stabilized Mechanization	20
3.5 Monte Carlo Analysis	22
4.0 RESULTS AND DISCUSSION	27
4.1 Trajectory Simulation	27
4.2 IMU Selection	27
4.3 Performance Metrics	28
4.3.1 Error Probables	29
4.3.2 Miss Distance and Terminal Error	29
4.3.3 Standard Deviation in Position and Attitude	30
5.0 CONCLUSIONS	38
REFERENCES.	39
APPENDIX - Calculating Allan Variance	40
LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS	41

LIST OF FIGURES

Figure	Page
1 Three Distributions Implemented for Error Parameters in SAIMUN	6
2 Accelerometer Output Corrupted by Startup Bias	6
3 Accelerometer Output Corrupted by Bias Instability	8
4 Accelerometer Output Corrupted by Thermo-Mechanical Zero-Mean White Noise	8
5 Three-Axis Accelerometer Output Corrupted by Scale Factor, Misalignment, and Nonorthog- onality Errors	10
6 Accelerometer Output Corrupted by Quantization Error	10
7 Underlying Noise Processes in Allan Variance Analysis	11
8 Allan Variance Analysis of Example Accelerometers	12
9 Allan Variance Analysis for the Baseline Accelerometer	12
10 Examples of Stable Table IMU Mechanizations	13
11 Representation of the NED Coordinate Frame	17
12 Input and Output Samples from the Monte Carlo Simulation	25
13 Distribution Contours Obtained by the Mean and Covariance from the Monte Carlo Sim- ulation and Linearization	25
14 Input and Output Samples from the Monte Carlo Simulation (modified problem)	26
15 Distribution Contours Obtained by the Mean and Covariance from the Monte Carlo Sim- ulation and Linearization (modified problem)	27
16 Miss Distance Histograms for Two- and Three-Dimensional Positions: Case 1	30
17 Miss Distance Histograms for Two- and Three-Dimensional Positions: Case 2	31
18 Terminal Position Error of the Monte Carlo Samples for the ADIS16488 IMU: Case 1	32
19 Terminal Position Error of the Monte Carlo Samples for the 3DM-GX3-15 IMU: Case 12	33
20 Terminal Position Error of the Monte Carlo Samples for the ADIS16488 IMU: Case 2	34
21 Terminal Position Error of the Monte Carlo Samples for the 3DM-GX3-15 IMU: Case 2	35
22 Position Standard Deviations as a Function of Time for the ADIS16488 and the 3DM- GX3-15 IMUs: Case 1	36
23 Attitude Standard Deviations as a Function of Time for the ADIS16488 and the 3DM- GX3-15 IMUs: Case 1	36
24 Position Standard Deviations as a Function of Time for the ADIS16488 and the 3DM- GX3-15 IMUs: Case 2	37
25 Attitude Standard Deviations as a Function of Time for the ADIS16488 and the 3DM- GX3-15 IMUs: Case 2	37

ACKNOWLEDGMENTS

This material is based on research sponsored by Air Force Research Laboratory under agreement number FA9453-13-1-0289. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

DISCLAIMER

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

(This page intentionally left blank)

1.0 SUMMARY

A primary aspect involved in the navigation of a wide variety of vehicles, such as ground vehicles and aircraft, is the use of inertial measurement units. The purpose of the inertial measurement unit is to provide an internal measure of the linear acceleration and angular velocity of the vehicle in order to be used by a navigation system in the dead-reckoning of the vehicle's position and attitude. Due to the nature of sensors, the measurement of the linear acceleration and angular velocity are corrupted by several error sources, such as bias and noise, amongst others. In order to assess the coupled impact of these error sources on the navigation accuracy for an arbitrary vehicle, a simulation and analysis tool is developed that can model several inertial measurement unit mechanizations, simulate the error sources encountered in the acquisition of measurement data, emulate the navigation software, and perform a range of statistical analyses via Monte Carlo simulation. The developed tool is applied to a ground vehicle navigation scenario in order to assess the performance of different sensors and provide conclusions as to when each sensor should be used to achieve the best navigation performance.

2.0 INTRODUCTION

Inertial measurement unit (IMU) technology is used to drive a dominant element of navigation systems on vehicles ranging from submarines to ground vehicles, missiles, and spacecraft. The purpose of the IMU is to provide an internal measure of the changes in the linear and angular motions of the vehicle due to non-gravitational stimuli by using triads of single-axis accelerometers to measure the linear motion and gyros to measure the angular motion via measurement of the angular velocity. In doing so, the position, velocity, and attitude of the vehicle can be predicted from a prescribed starting condition along with a representation of the confidence in the prediction, i.e. an associated uncertainty.

Traditionally, IMU-based navigation is mechanized in one of two ways: gimbaled/gyro-stabilized or strapdown. In the gimbaled mechanizations, the triad of accelerometers is placed on a "stable table" at the center of set of interconnected gimbals or at the center of a fluidic stabilization system. In the strapdown mechanization, the triad of accelerometers and gyros are fixed (strapped down) to the body of the vehicle, and there is no stable table. The different mechanizations effectively alter the way in which the triad orientation is handled: for gimbaled systems, mechanical tracking is used, whereas computational tracking is used for strapdown systems. Common to every system, the outputs of the single-axis sensors that make up the accelerometer and gyroscope are used in conjunction with the navigation system software to dead-reckon (i.e. using only the IMU) the state (position, velocity, and attitude) of the vehicle. Since the single-axis sensors are imperfect (corrupted by measurement errors), knowledge of the exact position, velocity, and attitude of the vehicle will degrade over time, which leads to navigation errors that may be described by a statistical distribution.

Typical analysis of IMU-driven systems employs limited statistical information regarding the distribution of the position, velocity, and attitude of the vehicle. Namely, the standard linear covariance analysis makes use of only the first two central moments of the distribution, i.e. the mean and covariance. The covariance obtained from linear covariance analysis is usually interpreted as being representative of a Gaussian distribution in order to determine performance measures; however, if the distribution is not Gaussian, the calculation of performance measures can lead to erroneous conclusions. Additionally, using only the mean and covariance information from a Gaussian distribution limits the type of performance analysis that can be conducted. Monte Carlo techniques present an attractive alternative that obviates erroneous statistical conclusions

obtained from a Gaussian assumption resulting from linear covariance analysis by working with a fully probabilistic representation of the navigation errors.

The following objectives are identified as necessary for the development of a tool that can simulate and analyze the performance of IMU-based navigation systems:

1. Formulate dynamical system representations of the state evolution model for gimbal- and strapdown-based systems.
2. Develop a general IMU model capable of simulating the corrupted acceleration and angular velocity of an IMU given true acceleration and angular velocity trajectories.
3. Identify relevant performance criteria to be assessed, such as,
 - error probables,
 - terminal miss distance Monte Carlo sample histograms,
 - terminal miss distance Monte Carlo sample clouds, and
 - position, velocity, and attitude standard deviation over time.
4. Build a simulation environment to assess navigation performance which contains:
 - a vehicle model (i.e. gimbal-based or strapdown-based vehicle),
 - an IMU error model that allows for user-selectable inclusion of different error sources and statistics of the parameterization,
 - an IMU-based dead-reckoning navigation model, and
 - an analysis model that allows for user-selectable performance criteria to be assessed.
5. Perform Monte Carlo analyses and assess performance criteria.

In order to quantify the effects of the IMU parameters on navigation error, SAIMUN (Simulation and Analysis of IMU-based Navigation) was developed. This tool utilizes a known truth trajectory, IMU parameterization of interest, and initial conditions to calculate the performance criteria. The tool makes use of Monte Carlo analysis to propagate the position, velocity, and attitude uncertainty forward in time. Each Monte Carlo sample is generated according to the initial uncertainty and is then dead-reckoned with corrupted acceleration and angular velocity data according to the IMU parameterization. Statistics are collected from these Monte Carlo samples over time to obtain the performance criteria.

SAIMUN was developed to consider three different IMU mechanizations: strapdown, space-stabilized, and local-level. The strapdown IMU mechanization relies on microelectromechanical systems (MEMS) technology for the accelerometer and gyroscope triads. The triads are stationary with respect to the IMU case; thus, the attitude is dead-reckoned numerically using the gyroscope data to provide the vehicle attitude. The space-stabilized and local-level IMU mechanizations rely on a mechanical platform to track a desired reference frame in which the accelerometer triad is oriented. The local-level mechanization tracks the North-East-Down coordinate frame; the space-stabilized mechanization tracks an inertial coordinate frame.

The remainder of this report is organized as follows. Section 3.1 provides an overview of the simulation environment that was developed. Section 3.2 details the modeling of the error sources which contribute to the IMU corruption and provides the overall models for the measured IMU outputs that are used in the strapdown, local-level, and space-stabilized implementations. In Section 3.3, an overview of the Allan variance analysis method and its utility for analyzing noise

processes present in IMU data is given, which is followed by a detailed development of the dynamical models used for the strapdown, local-level, and space-stabilized IMUs in Section 3.4. Section 3.5 describes the Monte Carlo analysis technique and its advantages over other uncertainty analysis techniques. Section 4.0 presents the results of a ground vehicle navigation simulation in which two IMUs are considered and compared using the developed tool. Section 5.0 concludes the report with a discussion as to how the developed tool is used to select an appropriate IMU for the ground vehicle navigation simulation as well as a discussion on other capabilities that SAIMUN offers.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 SAIMUN Overview

In order to analyze the effect of inertial measurement unit performance on navigation systems, SAIMUN was developed. SAIMUN utilizes Monte Carlo sampling to propagate the initial position, velocity, and attitude uncertainty forward in time, while accounting for the injection of uncertainty into the position, velocity, and attitude due to error corruption present in the IMU. The effect of the following IMU error sources on the position, velocity, and attitude uncertainty is quantified in SAIMUN:

- Startup bias
- Bias instability
- Random walk
- Scale factor error
- Axes misalignment
- Axes nonorthogonality
- Quantization

The modeling of these IMU error parameters is outlined in Section 3.2. In addition to information about the IMU error sources, the tool also requires the following information:

- Tabulated true trajectory, which is provided as an input to SAIMUN to ensure that the tool can be applied to any vehicle, ranging from ground vehicles to aquatic vehicles, air vehicles, and spacecraft. The tabulated truth trajectory consists of the following information:
 - inertial position vector of the center of mass (CM) of the vehicle,
 - inertial velocity vector of the CM of the vehicle,
 - inertial acceleration vector of the CM of the vehicle,
 - attitude of the vehicle body frame, expressed as a quaternion and defined as a 3-2-1 rotation from the inertial frame,
 - angular velocity of the vehicle body frame expressed in the vehicle body frame,
 - angular acceleration of the vehicle body frame expressed in the vehicle body frame, and
 - seconds elapsed since the beginning of the trajectory.
- Initial epoch, which corresponds to the start of the tabulated truth trajectory and is specified as a Julian Date.

- Free Navigation Drift time, which is the wait time for the vehicle between the specified initial epoch and the beginning of the trajectory specified by the tabulated truth data. The vehicle will begin navigating at the initial epoch, and is assumed fixed to the Earth’s surface until the Free Navigation Drift time has elapsed.
- Δ_{UTC}^{UT1} , which is the time offset between Coordinated Universal Time (UTC) and the raw uncorrected Universal Time (UT) corrected for polar wandering (UT1).
- IMU location aboard the vehicle with respect to the CM. SAIMUN synthesizes the centrifugal and Euler accelerations experienced by the IMU due to the angular velocity and angular acceleration of the vehicle.
- IMU timestep, which is the time between successive measurements of the acceleration and angular velocity provided by the IMU.
- IMU mechanization, which can be any one of the following:
 - strapdown,
 - space-stabilized, or
 - local-level.

The IMU mechanizations are further described in Section 3.4.

- Number of Monte Carlo sample trajectories. This parameter is a trade between computational effort required and the accuracy of the results.
- Covariance of the initial position, velocity, and attitude of the vehicle, corresponding to the initial epoch.

SAIMUN then implements Monte Carlo analysis to propagate the initial uncertainty forward in time and outputs the following plots:

- position standard deviation over time,
- velocity standard deviation over time,
- attitude standard deviation over time,
- three-dimensional terminal miss distance histogram of the Monte Carlo samples,
- two-dimensional terminal miss distance histogram of the Monte Carlo samples projected in the East-North plane,
- propagation error of the trajectory dead-reckoned with the uncorrupted acceleration and angular velocity, defined as the *propagated truth*, with respect to the tabulated true trajectory input, which is a measure of the numerical degradation due to discretization of the continuous-time equations of motion,
- three-dimensional point cloud of the terminal positions of the Monte Carlo sample trajectories in the East-North-Up (ENU) coordinate frame,
- two-dimensional point cloud of the terminal positions of the Monte Carlo sample trajectories projected in the East-North plane,

- two-dimensional point cloud of the terminal positions of the Monte Carlo sample trajectories projected in the Up-East plane,
- two-dimensional point cloud of the terminal positions of the Monte Carlo sample trajectories projected in the North-Up plane, and
- three-dimensional plot of the true and propagated truth trajectories.

SAIMUN also outputs the following information:

- Circular Error Probable (CEP), which is defined as the radius of a circle centered about the propagated true terminal position that contains half of the East-North projected terminal positions obtained from the Monte Carlo samples,
- Spherical Error Probable (SEP), which is defined as the radius of a sphere about the propagated true terminal position that contains half of the terminal positions obtained from the Monte Carlo samples, and
- Height Error Probable (HEP), which is defined as the altitude limits (defining an interval) centered about the propagated true terminal position that contains half of the terminal altitudes obtained from the Monte Carlo samples.

3.2 IMU Modeling

The acceleration and angular velocity measured by an IMU are corrupted by a variety of error sources. SAIMUN accounts for the most dominant sources of error by specifying a probability density function (pdf) for the error parameters to define each error source. The pdfs implemented in SAIMUN are

- the Dirac distribution, which is used to represent a constant parameter specified solely by the mean,
- the normal distribution, which represents a Gaussian distributed parameter that is solely defined by a mean and variance (or standard deviation), and
- the uniform distribution, which represents a parameter that is equiprobable between lower and upper bounds.

These distributions are graphically represented in Fig. (1).

The IMU model accounting for error sources is given for the accelerometers and gyroscopes as

$$\mathbf{a}_{m,k} = \mathbf{a}_{Q,k} \left((\mathbf{I} + \mathbf{N}_a + \mathbf{M}_a) (\mathbf{I} + \mathbf{S}_a) (\mathbf{a}_k + \mathbf{b}_{a,0} + \mathbf{b}_{a,k} + \mathbf{w}_{a,k}) - \mathbf{E} \{ \mathbf{b}_{a,0} \} \right) \quad (1a)$$

$$\boldsymbol{\omega}_{m,k} = \boldsymbol{\omega}_{Q,k} \left((\mathbf{I} + \mathbf{N}_g + \mathbf{M}_g) (\mathbf{I} + \mathbf{S}_g) (\boldsymbol{\omega}_k + \mathbf{b}_{g,0} + \mathbf{b}_{g,k} + \mathbf{w}_{g,k}) - \mathbf{E} \{ \mathbf{b}_{g,0} \} \right), \quad (1b)$$

where

\mathbf{a}_k is the true acceleration experienced by the IMU in the IMU frame at t_k ,

$\mathbf{b}_{a,0}$ is the startup bias of the accelerometers,

$\mathbf{b}_{a,k}$ is the bias of the accelerometer at t_k , which changes due to bias instability,

$\mathbf{w}_{a,k}$ is the thermo-mechanical zero-mean white noise present in the accelerometers,

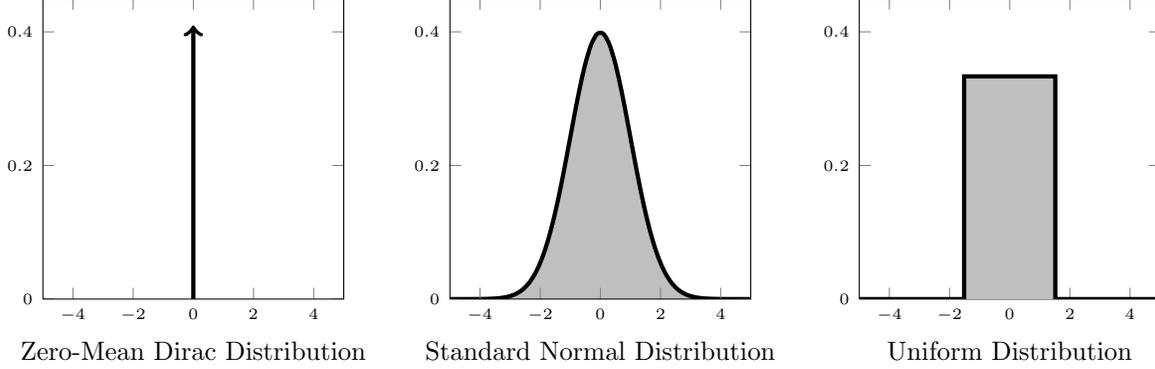


Figure 1. Three Distributions Implemented for Error Parameters in SAIMUN

S_a is the scale factor error matrix of the accelerometers,

M_a is the axes misalignment matrix of the accelerometers,

N_a is the axes nonorthogonality matrix of the accelerometers, and

$a_{Q,k}$ is the quantization effect caused by analog-to-digital conversion.

and similarly for the gyroscopes. The error sources will now be presented as they affect the accelerometer. The sources of error used by this model affect the gyroscopes identically and thus their presentation is omitted.

3.2.1 Startup Bias. The startup bias affecting the accelerometers, $b_{a,0}$, is the average output of the accelerometers while they are not undergoing any acceleration [1]. For practical applications, the startup bias is quantified by accounting for and removing the accelerometer output due to gravity while the IMU remains motionless. Figure (2) shows an example of the startup bias of an accelerometer undergoing no acceleration. SAIMUN assumes that the statistics of the startup bias

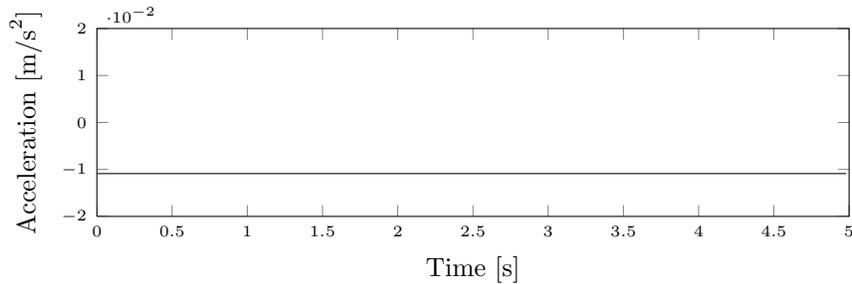


Figure 2. Accelerometer Output Corrupted by Startup Bias

are well known and can be approximated by one of the three supported error-parameter pdfs. The startup bias is then realized from the error-parameter pdf during virtual IMU initialization.

3.2.2 Bias Instability. The bias of an accelerometer tends to drift slowly over time and is known as a *walking* bias [1]. The walking bias affecting the accelerometers at t_k , $b_{a,k}$, is the change in the bias since the IMU was powered on. This effect is modeled by a first-order random walk

model, which causes a linear increase in the variance of the accelerometer bias with time. The discrete first-order random walk model is given by

$$\mathbf{b}_{a,k} = \mathbf{b}_{a,k-1} + \mathbf{w}_{\text{BI},k}, \quad (2)$$

where $\mathbf{w}_{\text{BI},k}$ is a zero-mean white noise process of standard deviation σ_{BI} . The initial standard deviation in the walking bias is zero, and the standard deviation at a later time, t_{BI}^* , is specified as σ_{BI}^* . Utilizing the zero initial condition and applying Eq. (2) recursively from t_0 to t_k yields the walking bias at t_k as

$$\mathbf{b}_{a,k} = \sum_{i=1}^{t_k/\Delta t} \mathbf{w}_{\text{BI},k}, \quad (3)$$

where Δt is the timestep of the IMU. Noting that the variance of the summation of n independent random variables denoted by X_i is equivalent to the sum of the variances of n independent random variables, i.e.

$$\text{Var} \left[\sum_{i=1}^n X_i \right] = \sum_{i=1}^n \text{Var} [X_i];$$

this yields

$$\text{Var} [\mathbf{b}_{a,k}] = \sum_{i=1}^{t_k/\Delta t} \text{Var} [\mathbf{w}_{\text{BI},k}] = \frac{t_k}{\Delta t} \sigma_{\text{BI}}^2, \quad (4)$$

when applied to Eq. (3). Evaluating Eq. (4) at the boundary condition gives

$$\frac{t_k^*}{\Delta t} \sigma_{\text{BI}}^2 = \sigma_{\text{BI}}^{*2}. \quad (5)$$

The standard deviation of the zero-mean white noise process driving the random walk model, σ_{BI} , can now be found by manipulating Eq. (5) as

$$\sigma_{\text{BI}} = \sqrt{\frac{\Delta t}{t_k^*} \sigma_{\text{BI}}^*}. \quad (6)$$

Figure (3) shows an example accelerometer undergoing a random walk due to bias instability. SAIMUN inputs the boundary conditions σ_{BI}^* and t_{BI}^* , the IMU timestep, Δt , and the desired error-parameter pdf-type for \mathbf{w}_k and calculates the standard deviation of the underlying noise process driving the random walk model according to Eqs. (2)–(6). Each step in the random walk model is realized independently for each IMU.

3.2.3 Velocity Random Walk. When thermo-mechanical zero-mean white noise, $\mathbf{w}_{a,k}$, present in the accelerometers is integrated, a random walk is produced in the velocity state, and the standard deviation of the velocity grows proportionally to the square-root of time [1]. The variance of the white noise present in the accelerometers is described by the Velocity Random Walk (VRW) specification. The standard deviation of the zero-mean white noise is given as a function of the VRW specification as

$$\sigma_{\text{VRW}} = \frac{\text{VRW}_{\text{spec}}}{\sqrt{\Delta t}}.$$

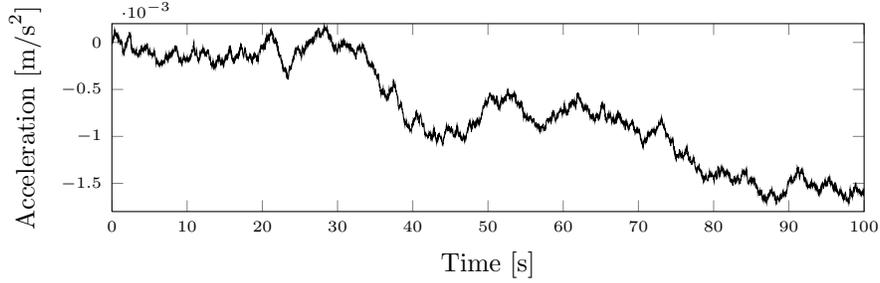


Figure 3. Accelerometer Output Corrupted by Bias Instability

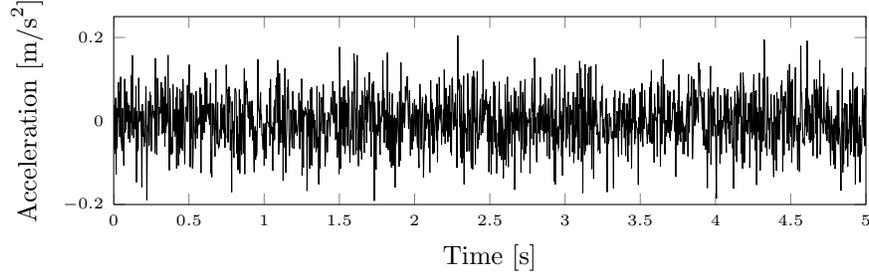


Figure 4. Accelerometer Output Corrupted by Thermo-Mechanical Zero-Mean White Noise

Example accelerometer output corrupted by zero-mean white-noise is shown in Fig. (4). SAIMUN inputs the VRW specification and IMU timestep to calculate the standard deviation of the white noise process underlying the velocity random walk. This zero-mean white noise process is realized independently at each t_k for the virtual IMU.

3.2.4 Scale Factor Error. Errors in analog-to-digital converters and other circuitry can induce scale factor errors in the accelerometers, denoted by \mathbf{S}_a . For instance, errors may be incurred due to reference voltage errors caused by a variety of sources, including, among other things, temperature fluctuations and calibration errors. For illustration purposes, consider a simple example of a generic scale factor error, s , and assume that the nominal reference voltage for the analog-to-digital converter is $V_{\text{nom}} = 5 \text{ V}$, while the actual reference voltage provided to the analog-to-digital converter is $V_{\text{act}} = 5.02 \text{ V}$. The scale factor error of the signal being converted by the analog-to-digital converter is then given by

$$s = \frac{V_{\text{nom}} - V_{\text{act}}}{V_{\text{act}}} = \frac{5 \text{ V} - 5.02 \text{ V}}{5.02 \text{ V}} = -0.003984 = -0.3984\%.$$

Denoting the scale factor errors for each axis of the accelerometer triad by s_x , s_y , and s_z , the total accelerometer scale factor error may be expressed as

$$\mathbf{S}_a = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}.$$

SAIMUN inputs an error-parameter pdf for s_x , s_y , s_z and realizes these parameters for the virtual IMU from this pdf.

3.2.5 Axes Misalignment Error. Errors in the manufacturing process and mounting of an IMU can lead to axes misalignment errors, denoted by \mathbf{M}_a . That is, the actual sensor axes are offset from the IMU case frame by a set of small rotations. This is readily seen by considering the term $\mathbf{I} + \mathbf{M}_a$ in Eq. (1a), which is of the form of a rotation matrix for small angles. Therefore, the actual sensed outputs of the IMU are not aligned perfectly with the modeled case frame of the IMU. The accelerometer misalignment errors are represented within \mathbf{M}_a by m_x , m_y , and m_z , with each term corresponding to its respective sensor axis. The misalignment errors are expressed in matrix form as

$$\mathbf{M}_a = \begin{bmatrix} 0 & m_z & -m_y \\ -m_z & 0 & m_x \\ m_y & -m_x & 0 \end{bmatrix}.$$

SAIMUN inputs an error-parameter pdf for m_x , m_y , m_z and realizes these parameters for the virtual IMU from this pdf.

3.2.6 Axes Nonorthogonality Error. Errors in the manufacturing process of an IMU can lead to axes nonorthogonality errors, denoted by \mathbf{N}_a . If the sensor had no nonorthogonality errors, each axis of the sensor triad would be at perfect right angles to the other two axes. Therefore, the nonorthogonality errors account for small deviations from perfectly orthogonal axes and allow for the modeling of sensor outputs that effectively measure small overlaps between the individual axes. The accelerometer axes nonorthogonality errors are represented within \mathbf{N}_a by n_x , n_y , and n_z , with each term corresponding to its respective sensor axis. The nonorthogonality errors can be expressed in matrix form as

$$\mathbf{N}_a = \begin{bmatrix} 0 & n_z & n_y \\ n_z & 0 & n_x \\ n_y & n_x & 0 \end{bmatrix}.$$

SAIMUN inputs an error-parameter pdf for n_x , n_y , n_z and realizes these parameters for the virtual IMU from this pdf.

Example output of an accelerometer triad undergoing 1 m/s acceleration in the z direction with scale factor, axes misalignment, and axes nonorthogonality errors is shown in Fig. (5).

3.2.7 Quantization Error. Quantization error arises in IMUs due to the limitations of analog-to-digital signal conversion and is modeled as affecting the measurements of acceleration and angular velocity as

$$\mathbf{a}_{Q,k}(\mathbf{a}_k) = \text{round} \left(\frac{2^{n-1}}{a_{\max}} \mathbf{a}_k \right) \frac{a_{\max}}{2^{n-1}},$$

where n is the bitrate of the analog to digital converter, and a_{\max} is the maximum value able to be quantized by the analog to digital converter. It is noted that the quantization error is applied after all of the other errors have been applied, such that the output of the quantization error is the measured acceleration of Eq. (1a). Values of the bitrate and maximum value are both input into SAIMUN. The quantization error in angular velocity is calculated similarly. Figure (6) shows a quantized signal from an IMU with thermo-mechanical noise present.

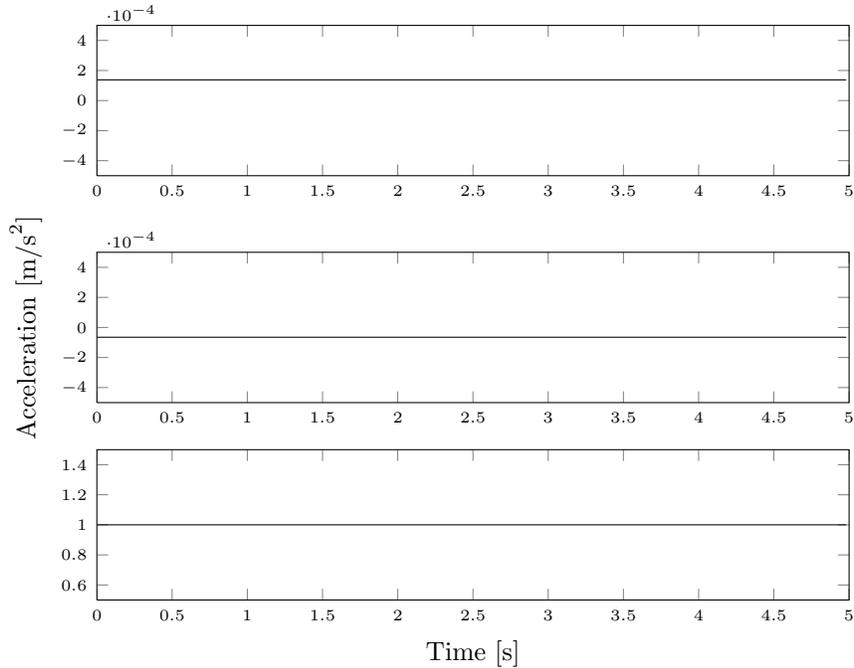


Figure 5. Three-Axis Accelerometer Output Corrupted by Scale Factor, Misalignment, and Nonorthogonality Errors

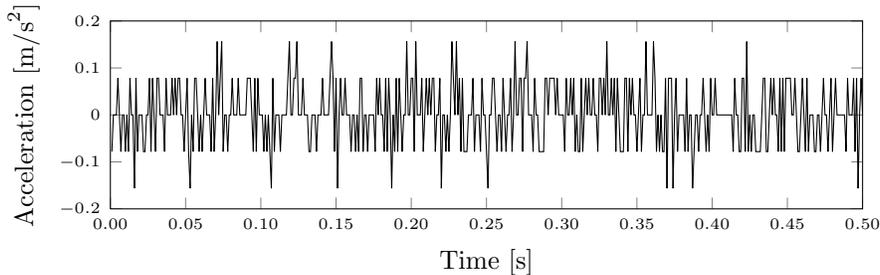


Figure 6. Accelerometer Output Corrupted by Quantization Error

3.3 Allan Variance

Section 3.2 outlined the modeling of the dominant error sources present in the measured acceleration (and equivalently the angular velocity) signals of an IMU. In order to utilize these models the necessary parameters must be quantified and input. One technique to quantify the parameters necessary to model the velocity random walk and the bias instability parameters is Allan variance analysis [1, 2, 3]. Allan variance analysis quantifies the variance between *averaging times* of a signal in order to determine its underlying noise process. A typical Allan variance plot is shown in Fig. (7).

The slope of the Allan variance curve on a log-log plot corresponds to the type of random process dominating the signal over the given averaging time. The slope, m , of the Allan deviation (the square root of the Allan variance) curve for different underlying noise processes present in an IMU signal consists of the following:

- $m = -0.5$ corresponds to a white noise process,

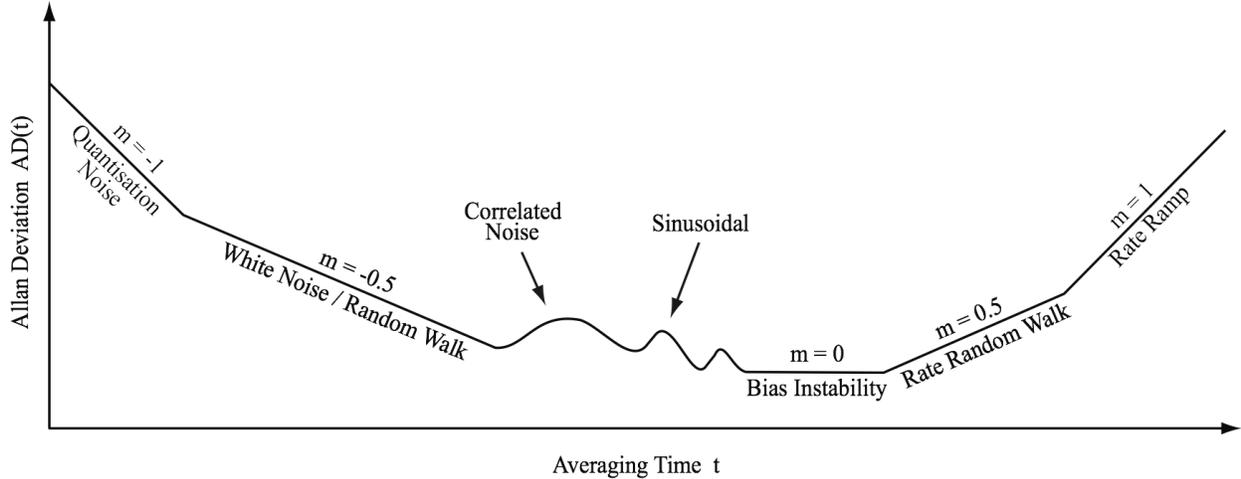


Figure 7. Underlying Noise Processes in Allan Variance Analysis [3]

- $m = 0$ corresponds to bias instability, and
- $m = 0.5$ corresponds to a random walk process.

To illustrate the slopes corresponding to each of the aforementioned noise processes, an example of computing the Allan deviation for synthetic accelerometer is considered. Three different accelerometers are investigated: a baseline accelerometer, an accelerometer with double the velocity random walk specification (relative to the baseline), and an accelerometer with double the bias instability specification (relative to the baseline). Data on each accelerometer is simulated for 1 day at a frequency of 100 Hz. This data is then used to construct the Allan deviation plot given in Fig. (8) using the algorithm outlined in the Appendix. After these plots are generated for the accelerometers, the curves can be used to extract the velocity random walk and bias instability specifications [1].

Comparison of the baseline and double VRW Allan deviation curves shows that an increase in white noise caused by the increased VRW specification causes an upward vertical shift in the portion of the curves representing white noise, which has $m = -0.5$. Because the bias instability specifications are identical for the baseline and double VRW curves, the portion of the Allan deviation curves representing the random walk, which has $m = 0.5$, are approximately collinear.

Comparison of the baseline and double bias instability Allan deviation curves shows that an increase in bias instability causes an upward vertical shift in the portion of the curves representing random walk, which has $m = 0.5$. Because the VRW specifications are identical for the baseline and double bias instability curves, the portion of the Allan deviation curves representing the white noise, which has $m = -0.5$ are approximately collinear.

The velocity random walk specification, VRW_{spec} , is determined by the intersection of a line fitted to the portion of the Allan deviation curve corresponding to the white noise process ($m = -0.5$) with an averaging time of one second. The intersection point determining VRW_{spec} for the previously described baseline accelerometer is illustrated by the blue marker in Fig. (9). Table 3 summarizes this specification for the baseline accelerometer. The bias instability parameters, t_k^* and σ_{BI}^* , are determined by the minimum point of the Allan deviation curve. This point is shown by the green marker for the baseline accelerometer in Fig. (9), and the specifications are summarized for the baseline accelerometer in Table 1.

Utilizing Allan variance analysis allows for the measured acceleration and angular velocity from

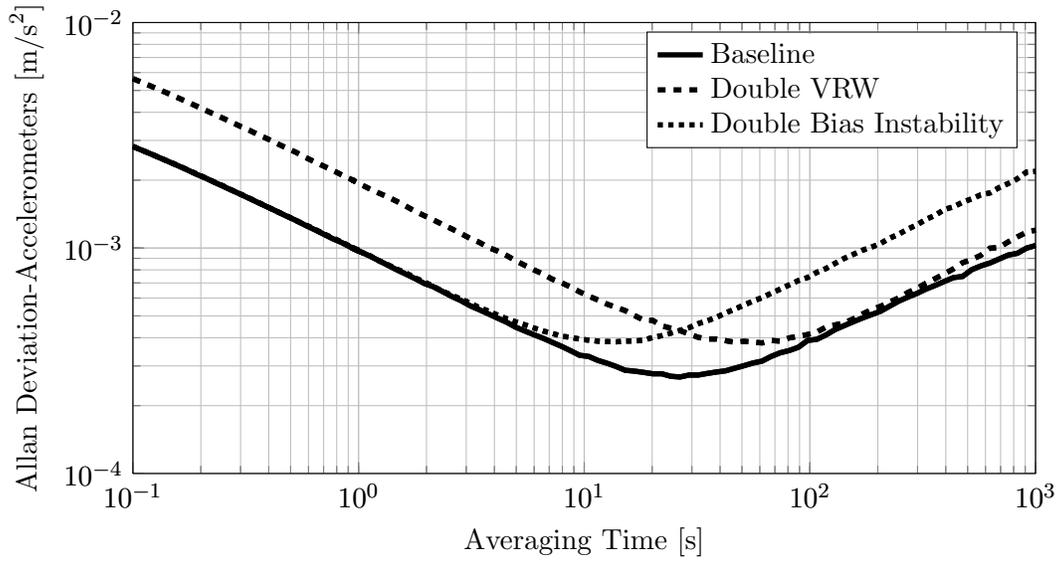


Figure 8. Allan Variance Analysis of Example Accelerometers

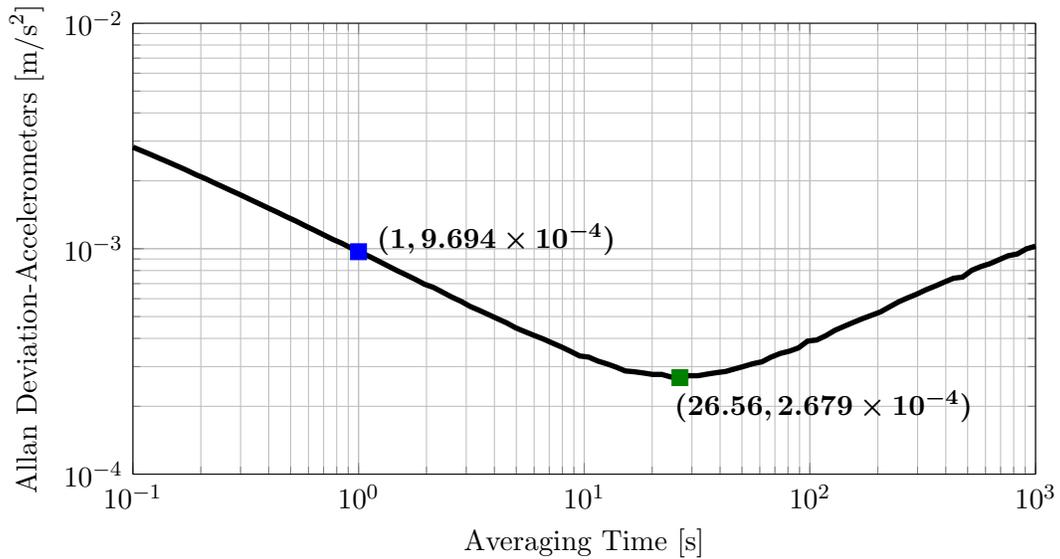


Figure 9. Allan Variance Analysis for the Baseline Accelerometer

Table 1. VRW and Bias Instability Specifications from Allan Variance Analysis for the Baseline Accelerometer

Specification	Value
VRW_{spec}	$9.694 \times 10^{-4} \frac{m}{s^2}/\sqrt{s}$
t_k^*	26.56 s
σ_{BI}^*	$2.679 \times 10^{-4} \frac{m}{s^2}$

different IMUs to be compared and contrasted through trends in the Allan deviation curves. While the above discussion has focused on Allan variance analysis as applied to accelerometers, the same process can be used when considering gyroscopes.

3.4 IMU Mechanizations

Traditionally, IMU-based navigation is mechanized in one of two ways: gimbaled/gyro-stabilized or strapdown. In the gimbaled mechanization, the triad of accelerometers is placed on a “stable table” at the center of set of interconnected gimbals. This enables the stable table, through the use of feedback tracking using the gyros, to track a predetermined attitude such that the accelerometers measure a certain, desirable resultant linear motion. Typically, the stable table is configured to track a north-east-down coordinate system, in which case the accelerometers measure the linear motion in these directions, or it is configured to track the inertial coordinate system (i.e. to remain fixed in space as the vehicle maneuvers).

The most basic of gimbal-based mechanizations employs three interconnected gimbal rings, such as is found on the Minuteman III intercontinental ballistic missile (see Fig. (10(a))). While this configuration provides the ability to actively control the orientation of the stable table through motors mounted on the gimbals, the use of three gimbals sometimes leads to gimbal lock and prevents the stable table from tracking all possible attitudes that may be required.



Figure 10. Examples of Stable Table IMU Mechanizations [4]

More advanced configurations of the stable table mounting design can be used to replace the original gimbaled system. This is accomplished by using four or more gimbals or by using a fluid

stabilization system, which has the stable table floating in the center of a sphere filled with pressurized fluid. Whereas the attitude of the stable table is controlled using torquers and manipulation of the gimbals in the original design, the attitude of the stable table in the fluidic design is controlled via nozzle actuators which avoids any possible complications due to gimbal lock. One such implementation of the fluidic system is the Advanced Inertial Reference Sphere (AIRS) that was developed for the LGM-118A Peacekeeper (see Fig. (10(b))).

In the strapdown mechanization, the triads of accelerometers and gyros are fixed (strapped down) to the body of the vehicle, and there is no stable table. This means that where the stable table was used to mechanically track the inertial frame in order to track a predetermined attitude and sense accelerations in this rotating frame, the gyro outputs are now used to numerically track the orientation of the IMU (called the case frame) with respect to the inertial frame. To avoid any singularities present in three-parameter attitude representations such as Euler angles, the attitude quaternion is used and can be propagated forward in time using the sensed angular velocity from the IMU. Additionally, the accelerations are now sensed in the case frame, and these must be transformed using the case-to-inertial attitude in order to predict the translational position and velocity of the vehicle.

Three inertial navigation system mechanizations are presented here: a strapdown system, an actively commanded local-level gimbaled system, and a passively commanded inertially stable space-stabilized gimbaled system. For each of the aforementioned mechanizations, the governing equations for the position, velocity, and attitude of the IMU are presented, and a set of discretized equations are developed. The discretized equations governing the position, velocity, and attitude of the IMU then are representative of the software and hardware that would be used in each inertial navigation mechanization.

3.4.1 Strapdown Mechanization. The inertial position, \mathbf{r} , velocity, \mathbf{v} , and attitude, $\bar{\mathbf{q}}$, of a strapdown IMU aboard a vehicle are governed by [5, 6]

$$\begin{aligned}\dot{\mathbf{r}}_{\text{IMU}}^i(t) &= \mathbf{v}_{\text{IMU}}^i(t) \\ \dot{\mathbf{v}}_{\text{IMU}}^i(t) &= \mathbf{a}_g^i(\mathbf{r}_{\text{IMU}}^i(t) + \mathbf{T}_c^i(t)\mathbf{r}_{\text{CM/IMU}}^c) + \mathbf{T}_c^i(t)\mathbf{a}_{ng}^c(t) \\ \dot{\bar{\mathbf{q}}}_i^c(t) &= \frac{1}{2}\bar{\boldsymbol{\omega}}_{c/i}^c(t) \otimes \bar{\mathbf{q}}_i^c(t),\end{aligned}$$

where \mathbf{a} is acceleration, $\boldsymbol{\omega}$ is angular velocity, and \mathbf{T}_c^i is a rotation matrix that rotates a vector representation from the c to the i coordinate frame. $\bar{\boldsymbol{\omega}}$ represents the angular velocity expressed as a pure quaternion. The c/i subscript notation represents the c frame with respect to the i frame. The sub/superscripts refer to

IMU	inertial measurement unit
CM	center-of-mass of the vehicle
c	case frame of the IMU
i	Earth Centered Inertial frame
g	due to gravity
ng	not due to gravity

where the superscript for a vector quantity denotes the coordinate frame in which the vector is expressed. To simplify the nomenclature, let

$$\begin{aligned}\mathbf{r}_{\text{IMU}}^i(t) &\rightarrow \mathbf{r}(t), & \mathbf{v}_{\text{IMU}}^i(t) &\rightarrow \mathbf{v}(t), & \mathbf{a}_g^i(\cdot) &\rightarrow \mathbf{g}(\cdot), \\ \mathbf{T}_c^i(t) &\rightarrow \mathbf{T}^T(t), & \mathbf{a}_{ng}^c(t) &\rightarrow \mathbf{a}(t), & \bar{\mathbf{q}}_i^c(t) &\rightarrow \bar{\mathbf{q}}(t), \\ \boldsymbol{\omega}_{c/i}^c(t) &\rightarrow \boldsymbol{\omega}(t), & \mathbf{r}_{\text{CM/IMU}}^c &\rightarrow \mathbf{d}, & \text{and } \mathbf{r}_{\text{IMU}}^i(t) + \mathbf{T}_c^i(t)\mathbf{r}_{\text{CM/IMU}}^c &\rightarrow \mathbf{s}(t).\end{aligned}$$

With these substitutions, the equations of motion may be written succinctly as

$$\begin{aligned}\dot{\mathbf{r}}(t) &= \mathbf{v}(t) \\ \dot{\mathbf{v}}(t) &= \mathbf{g}(\mathbf{s}(t)) + \mathbf{T}^T(t)\mathbf{a}(t) \\ \dot{\bar{\mathbf{q}}}(t) &= \frac{1}{2}\bar{\boldsymbol{\omega}}(t) \otimes \bar{\mathbf{q}}(t).\end{aligned}$$

The continuous time strapdown IMU equations of motion are now discretized via analytical integration. In the development of these discretized equations of motion, it is assumed that the non-gravitational acceleration and angular velocity are constant over a small time-step, such that

$$\mathbf{a}_{m,k} = \frac{\Delta \mathbf{v}_{m,k}}{\Delta t_k} \quad \text{and} \quad \boldsymbol{\omega}_{m,k} = \frac{\Delta \boldsymbol{\theta}_{m,k}}{\Delta t_k}.$$

The continuous time attitude evolves as

$$\dot{\bar{\mathbf{q}}}(t) = \frac{1}{2}\bar{\boldsymbol{\omega}}_m(t) \otimes \bar{\mathbf{q}}(t),$$

where $\bar{\boldsymbol{\omega}}_m(t)$ is the pure quaternion given (in right-handed, vector-first form) by

$$\bar{\boldsymbol{\omega}}_m(t) = \begin{bmatrix} \boldsymbol{\omega}_m(t) \\ 0 \end{bmatrix}.$$

Define the quaternion $\Delta \bar{\mathbf{q}}(t)$ representing the rotation from the *a priori* attitude as

$$\Delta \bar{\mathbf{q}}(t) \triangleq \bar{\mathbf{q}}(t) \otimes \bar{\mathbf{q}}_{k-1}^{-1}.$$

Then, solving for $\bar{\mathbf{q}}(t)$ yields

$$\bar{\mathbf{q}}(t) \triangleq \Delta \bar{\mathbf{q}}(t) \otimes \bar{\mathbf{q}}_{k-1}.$$

Assuming the time step to be small, it can be shown that the rotation vector from t_{k-1} to t_k is

$$\boldsymbol{\theta}(t_k) = \boldsymbol{\omega}_{m,k} (t_k - t_{k-1}) = \boldsymbol{\omega}_{m,k} \Delta t_k,$$

where $\Delta t_k = t_k - t_{k-1}$. Thus, the discrete quaternion propagation is given by

$$\bar{\mathbf{q}}_k = \bar{\mathbf{q}}(t_k) = \bar{\mathbf{q}}(\boldsymbol{\omega}_{m,k} \Delta t_k) \otimes \bar{\mathbf{q}}_{k-1}, \quad (7)$$

where the quaternion representation of the arbitrary rotation vector $\boldsymbol{\theta}$ is given by

$$\bar{\mathbf{q}}(\boldsymbol{\theta}) = \begin{bmatrix} \sin\left(\frac{1}{2}\theta\right) \boldsymbol{\theta}/\theta \\ \cos\left(\frac{1}{2}\theta\right) \end{bmatrix}.$$

The continuous time velocity evolves as

$$\dot{\mathbf{v}}(t) = \mathbf{g}(\mathbf{s}(t)) + \mathbf{T}^T(t)\mathbf{a}_m(t).$$

Assume a small time step such that the non-gravitational acceleration is approximately constant, i.e. $\mathbf{a}_m(t) = \mathbf{a}_{m,k}$. Then, it follows that

$$\dot{\mathbf{v}}(t) = \mathbf{g}(\mathbf{s}(t)) + \mathbf{T}^T(t)\mathbf{a}_{m,k}. \quad (8)$$

The gravitational acceleration may be expanded in a first-order Taylor series expansion about the *a priori*, yielding

$$\mathbf{g}(\mathbf{s}(t)) = \mathbf{g}(\mathbf{s}_{k-1}) + \mathbf{G}_{k-1} (\mathbf{s}(t) - \mathbf{s}_{k-1}) ,$$

where

$$\mathbf{s}_{k-1} = \mathbf{r}_{k-1} + \mathbf{T}_{k-1}^T \mathbf{d}$$

and

$$\mathbf{G}_{k-1} = \left[\frac{\partial \mathbf{g}(\mathbf{s})}{\partial \mathbf{s}} \Big|_{\mathbf{s}=\mathbf{s}_{k-1}} \right] .$$

From the relationship that $\mathbf{s} = \mathbf{r} + \mathbf{T}^T \mathbf{d}$, and letting $\mathbf{r}(t) \rightarrow \mathbf{r}_{k-1}$, it follows that

$$\mathbf{g}(\mathbf{s}(t)) = \mathbf{g}(\mathbf{s}_{k-1}) + \mathbf{G}_{k-1} (\mathbf{T}^T(t) - \mathbf{T}_{k-1}^T) \mathbf{d} . \quad (9)$$

The first-order evolution of the rotation matrix is given by

$$\mathbf{T}(t) = \mathbf{T}_{k-1} - [\boldsymbol{\theta}(t) \times] \mathbf{T}_{k-1} ,$$

which may be transposed to give

$$\mathbf{T}^T(t) = \mathbf{T}_{k-1}^T + \mathbf{T}_{k-1}^T [\boldsymbol{\theta}(t) \times] . \quad (10)$$

Applying this relationship to the gravitational acceleration in Eq. (9), noting that $\boldsymbol{\theta}(t) = \boldsymbol{\omega}_{m,k}(t - t_{k-1})$, and defining $\mathbf{g}_{k-1} \triangleq \mathbf{g}(\mathbf{s}_{k-1})$, yields

$$\mathbf{g}(\mathbf{s}(t)) = \mathbf{g}_{k-1} - \mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] \boldsymbol{\omega}_{m,k}(t - t_{k-1}) . \quad (11)$$

Substituting Eqs. (10) and (11) into Eq. (8), simplifying, and rearranging terms yields

$$\dot{\mathbf{v}}(t) = \mathbf{g}_{k-1} + \mathbf{T}_{k-1}^T \mathbf{a}_{m,k} - (\mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] + \mathbf{T}_{k-1}^T [\mathbf{a}_{m,k} \times]) \boldsymbol{\omega}_{m,k}(t - t_{k-1}) . \quad (12)$$

Integrating Eq. (12) twice and once, respectively, yields the position and velocity evolution in time as

$$\begin{aligned} \mathbf{r}(t) &= \mathbf{r}_{k-1} + \mathbf{v}_{k-1}(t - t_{k-1}) + \frac{1}{2} \mathbf{g}_{k-1}(t - t_{k-1})^2 + \frac{1}{2} \mathbf{T}_{k-1}^T \mathbf{a}_{m,k}(t - t_{k-1})^2 \\ &\quad - \frac{1}{6} (\mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] + \mathbf{T}_{k-1}^T [\mathbf{a}_{m,k} \times]) \boldsymbol{\omega}_{m,k}(t - t_{k-1})^3 \\ \mathbf{v}(t) &= \mathbf{v}_{k-1} + \mathbf{g}_{k-1}(t - t_{k-1}) + \mathbf{T}_{k-1}^T \mathbf{a}_{m,k}(t - t_{k-1}) \\ &\quad - \frac{1}{2} (\mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] + \mathbf{T}_{k-1}^T [\mathbf{a}_{m,k} \times]) \boldsymbol{\omega}_{m,k}(t - t_{k-1})^2 . \end{aligned}$$

Letting $t \rightarrow t_k$ yields the discrete propagation equations for the position and velocity of the IMU as

$$\mathbf{r}_k = \mathbf{r}_{k-1} + \mathbf{v}_{k-1} \Delta t_k + \frac{1}{2} \mathbf{g}_{k-1} \Delta t_k^2 + \frac{1}{2} \mathbf{T}_{k-1}^T \mathbf{a}_{m,k} \Delta t_k^2 \quad (13a)$$

$$- \frac{1}{6} (\mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] + \mathbf{T}_{k-1}^T [\mathbf{a}_{m,k} \times]) \boldsymbol{\omega}_{m,k} \Delta t_k^3$$

$$\mathbf{v}_k = \mathbf{v}_{k-1} + \mathbf{g}_{k-1} \Delta t_k + \mathbf{T}_{k-1}^T \mathbf{a}_{m,k} \Delta t_k - \frac{1}{2} (\mathbf{G}_{k-1} \mathbf{T}_{k-1}^T [\mathbf{d} \times] + \mathbf{T}_{k-1}^T [\mathbf{a}_{m,k} \times]) \boldsymbol{\omega}_{m,k} \Delta t_k^2 . \quad (13b)$$

In summary, the discrete-time dead-reckoning equations for the position, velocity, and attitude of a vehicle with a strapdown IMU are given by Eqs. (13) and (7).

3.4.2 Local-Level Mechanization. The local-level mechanization physically commands the IMU platform via torquers into alignment with the local geographic frame tracking north, east, and down (NED). This class of navigation system maintains two of its accelerometers along the local horizon with the third aligned normal to the reference ellipsoid. To keep this alignment, the system rotates at an angular velocity equal to the Earth's rotation rate and the vehicle motion along the Earth and is independent of the vehicle body attitude. The navigational quantities of interest are the IMU frame position and Earth-referenced velocity vectors. A representation of the NED coordinate frame is shown in Fig. (11).

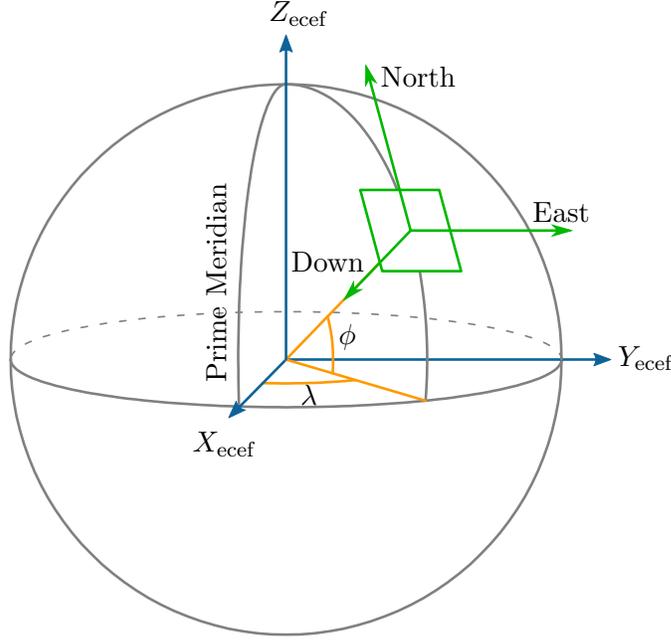


Figure 11. Representation of the NED Coordinate Frame¹

The geodetic latitude, longitude, and negative altitude are governed by [7]:

$$\ddot{\phi} = \frac{1}{(r_L + h)} \left\{ f_N - (\dot{r}_L + 2\dot{h})\dot{\phi} - \frac{1}{2}(r_l + h)(\dot{\lambda} + 2\omega_{ie})\dot{\lambda} \sin(2\phi) + \xi_g \right\} \quad (14a)$$

$$\ddot{\lambda} = \frac{1}{(r_l + h)} \left\{ \frac{f_E}{\cos \phi} + (r_L + r_l + 2h)\dot{\phi}\dot{\lambda} \tan \phi + 2(r_L + h)\dot{\phi}\omega_{ie} \tan \phi - 2\dot{h}(\dot{\lambda} + \omega_{ie}) - \dot{r}_l\dot{\lambda} - \frac{\eta_g}{\cos \phi} \right\} \quad (14b)$$

$$-\ddot{h} = f_D - (r_l + h)(\dot{\lambda} + 2\omega_{ie})\dot{\lambda} \cos^2 \phi - (r_L + h)\dot{\phi}^2 + g, \quad (14c)$$

where ϕ is the geodetic latitude, λ is the longitude, h is the altitude, ω_{ie} is the scalar rotation rate of the Earth about its spin axis. The terms f_N , f_E , and f_D are components of the specific force vector, \mathbf{f}_{sf} , which describe the values of the specific force in the north, east, and down directions respectively. These are the values of the sensed IMU acceleration in the local-level mechanization.

The magnitude of the geocentric position vector to the point on the ellipsoid directly below the vehicle is denoted by r_0 . The meridional radius of curvature, r_L , is the ellipsoid radius of curvature

¹Image modified with permission from http://en.wikipedia.org/wiki/File:ECEF_ENU_Longitude_Latitude_relationships.svg

in the north-south direction and the prime radius of curvature, r_l , is normal to the meridional radius of curvature in the east-west direction on the ellipsoid. The equations for computing r_0 , r_L , and r_l are

$$\begin{aligned} r_0 &= r_e \left[1 - \frac{1}{2} f (1 - \cos(2\phi)) + \frac{5}{16} f^2 (1 - \cos(4\phi)) \right] \\ r_L &= r_0 \left[1 - 2f \cos(2\phi) + \frac{1}{4} f^2 (1 - 4 \cos(2\phi) + 7 \cos(4\phi)) \right] \\ r_l &= r_0 \left[1 + 2f \sin^2 \phi - 3f^2 \sin^2 \phi (1 - 2 \sin^2 \phi) \right]. \end{aligned}$$

From the equations for r_0 , r_L , and r_l , temporal derivatives of the meridional radius of curvature and the prime radius of curvature can be determined. Furthermore, the evaluation of the radii depends upon r_e and f , which are, respectively, the equatorial radius and the flattening of the Earth ellipsoid. The flattening factor, f , is a measure of the compression between the equatorial and polar radii and is defined to be

$$f = \frac{r_e - r_p}{r_e},$$

where r_p is the polar radius of the Earth. The three quantities r_e , r_p , and f are constant for the Earth reference ellipsoid; for instance, when using the 1984 World Geodetic System (WGS-84) model for the reference ellipsoid, the parameters are given as shown in Table 2 (note that r_p is a derived constant).

Table 2. Parameters Defining the WGS-84 Model (adapted from [8])

Parameter	Value
r_e	6378.1370 km
r_p	6356.7523 km
f	1/298.257223563
ω_{ie}	7.2921150×10^{-5} rad/s
μ	3.986004418×10^5 km ³ /s ²

The final elements required to complete the evaluation of Eqs. (14) are the components of the gravity vector, ξ_g , η_g , and g . These are the components of the gravity vector referenced in the geographic (local-level) frame, which is given by

$$\mathbf{g}^n = \begin{bmatrix} \xi_g \\ -\eta_g \\ g \end{bmatrix} = \begin{bmatrix} \xi_g \\ -\eta_g \\ g_e + \Delta g \end{bmatrix} = \begin{bmatrix} G_N - r\omega_{ie}^2 \sin \phi \cos \phi_c \\ G_E \\ G_D - r\omega_{ie}^2 \cos \phi \cos \phi_c \end{bmatrix},$$

where ϕ_c is the geocentric latitude, ξ_g is the gravitational component due to meridian deflection of the vertical, η_g is the gravitation acceleration due to prime deflection of the vertical, g_e is the gravity magnitude associated with the reference ellipsoid, and Δg is the gravity anomaly, which is the difference between the observed gravitational acceleration and the value predicted from a given model. Values of the gravity anomaly are available online from the National Geospatial-Intelligence Agency. The geocentric latitude, ϕ_c , is defined as the angle between the equatorial plane and the normal and is given by the difference between the geodetic latitude and the deviation of the normal via

$$\phi_c = \phi - f(1 - h/r_0) \sin(2\phi).$$

To complete the evaluation of ξ_g , η_g , and g , G_N , G_E , and G_D are approximated as

$$\begin{aligned} G_N &\approx -G_\phi - G_r f \sin(2\phi) \\ G_E &= 0 \\ G_D &\approx -G_r + G_\phi f \sin(2\phi), \end{aligned}$$

where G_r and G_ϕ are the radial and colatitude components of the gravitational field in spherical coordinates. Retaining oblateness effects up to and including the fourth order terms, G_r and G_ϕ may be written as

$$\begin{aligned} G_r &= -\frac{\mu}{r^2} \left[1 - \frac{3}{2} J_2 \left(\frac{r_e}{r} \right)^2 (3 \cos^2 \varphi - 1) - 2 J_3 \left(\frac{r_e}{r} \right)^3 \cos \varphi (5 \cos^2 \varphi - 3) \right. \\ &\quad \left. - \frac{5}{8} J_4 \left(\frac{r_e}{r} \right)^4 (35 \cos^4 \varphi - 30 \cos^2 \varphi + 3) \right] \\ G_\phi &= 3 \frac{\mu}{r^2} \left(\frac{r_e}{r} \right)^2 \sin \varphi \cos \varphi \left[J_2 + \frac{1}{2} J_3 \left(\frac{r_e}{r} \right) \sec \varphi (5 \cos^2 \varphi - 1) + \frac{5}{6} J_4 \left(\frac{r_e}{r} \right)^2 (7 \cos^2 \varphi - 3) \right], \end{aligned}$$

where φ is the vehicle's colatitude.

The discrete-time form of the latitude, longitude, and negative altitude dynamics is formed using an Euler-integration method on the continuous-time dynamics. Defining a "position," "velocity," and "acceleration" to be

$$\mathbf{r} = \begin{bmatrix} \phi \\ \lambda \\ -h \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \dot{\phi} \\ \dot{\lambda} \\ -\dot{h} \end{bmatrix}, \quad \text{and} \quad \mathbf{a} = \begin{bmatrix} \ddot{\phi} \\ \ddot{\lambda} \\ -\ddot{h} \end{bmatrix},$$

the discrete propagation equations are given by

$$\mathbf{r}_k = \mathbf{r}_{k-1} + \mathbf{v}_{k-1} \Delta t_k \tag{15a}$$

$$\mathbf{v}_k = \mathbf{v}_{k-1} + \mathbf{a}_{k-1} \Delta t_k, \tag{15b}$$

where $\ddot{\phi}$, $\ddot{\lambda}$, and $-\ddot{h}$ are given by the evaluation of Eqs. (14).

In the local-level IMU mechanization, the stable table tracks the local-level (NED) coordinate system, which changes orientation as the vehicle's latitude and longitude change. The orientation of the local-level coordinate frame in reference to the inertial frame is given by the direction cosine matrix

$$\mathbf{T}_n^i = \begin{bmatrix} -\sin \phi \cos \tilde{\lambda} & -\sin \tilde{\lambda} & -\cos \phi \cos \tilde{\lambda} \\ -\sin \phi \sin \tilde{\lambda} & \cos \tilde{\lambda} & -\cos \phi \sin \tilde{\lambda} \\ \cos \phi & 0 & -\sin \phi \end{bmatrix},$$

where $\tilde{\lambda}$ is the celestial longitude, given by

$$\tilde{\lambda} = \lambda - \lambda_0 + \omega_{ie} t,$$

where λ is the terrestrial longitude from Greenwich, λ_0 is the initial terrestrial longitude, and t is the elapsed time. The angular velocity of the local-level frame is

$$\boldsymbol{\omega}_{i/n}^n = \begin{bmatrix} \dot{\tilde{\lambda}} \cos \phi \\ -\dot{\phi} \\ -\dot{\tilde{\lambda}} \sin \phi \end{bmatrix},$$

which may be used to form the time-wise evolution of the orientation of the local-level frame via [5]

$$\dot{\bar{\mathbf{q}}}_n^i(t) = \frac{1}{2} \bar{\boldsymbol{\omega}}_{i/n}^n(t) \otimes \bar{\mathbf{q}}_n^i(t),$$

where $\bar{\boldsymbol{\omega}}_{i/n}^n(t)$ is the pure quaternion (in right-handed, vector-first) form of the local-level frame angular velocity, defined as

$$\bar{\boldsymbol{\omega}}_{i/n}^n(t) = \begin{bmatrix} \boldsymbol{\omega}_{i/n}^n(t) \\ 0 \end{bmatrix}.$$

For notational ease, let $\boldsymbol{\omega}_{i/n}^n(t) \rightarrow \boldsymbol{\omega}(t)$ and $\bar{\mathbf{q}}_n^i(t) \rightarrow \bar{\mathbf{q}}(t)$.

The discrete evolution of the IMU platform attitude is as follows. Define the quaternion $\Delta\bar{\mathbf{q}}(t)$ representing the rotation from the *a priori* attitude as

$$\Delta\bar{\mathbf{q}}(t) \triangleq \bar{\mathbf{q}}(t) \otimes \bar{\mathbf{q}}_{k-1}^{-1}.$$

Solving for $\bar{\mathbf{q}}(t)$ yields

$$\bar{\mathbf{q}}(t) \triangleq \Delta\bar{\mathbf{q}}(t) \otimes \bar{\mathbf{q}}_{k-1}.$$

Assuming the time step to be small, it can be shown that the rotation vector from t_{k-1} to t_k is

$$\boldsymbol{\theta}(t_k) = \boldsymbol{\omega}_k (t_k - t_{k-1}) = \boldsymbol{\omega}_k \Delta t_k,$$

where $\Delta t_k = t_k - t_{k-1}$ and $\boldsymbol{\omega}_k = \boldsymbol{\omega}_{i/n}^n(t_k)$. Thus, the discrete quaternion propagation is given by

$$\bar{\mathbf{q}}_k = \bar{\mathbf{q}}(t_k) = \bar{\mathbf{q}}(\boldsymbol{\omega}_k \Delta t_k) \otimes \bar{\mathbf{q}}_{k-1}, \quad (16)$$

where the quaternion representation of the arbitrary rotation vector $\boldsymbol{\theta}$ is given by

$$\bar{\mathbf{q}}(\boldsymbol{\theta}) = \begin{bmatrix} \sin\left(\frac{1}{2}\theta\right) \boldsymbol{\theta}/\theta \\ \cos\left(\frac{1}{2}\theta\right) \end{bmatrix}.$$

In summary, the discrete time evolution of the local-level position, velocity, and IMU platform attitude are governed by Eqs. (15) and (16).

3.4.3 Space-Stabilized Mechanization. The space-stabilized mechanization tracks an Earth-centered, inertially nonrotating coordinate frame. This reference frame is realized by a gyro-stabilized platform with triaxial control. Except for torques applied as compensation to account for anisoelastic effects, thermal and acceleration sensitivity, etc., the gyroscopes are uncommanded and the system is “free-floating.”

The Earth-centered inertial acceleration of the IMU is taken as [7]

$$\ddot{\mathbf{r}}^i(t) = \mathbf{f}_m^i(t) + \mathbf{g}^i(\mathbf{r}^i(t)),$$

where \mathbf{r}^i is the inertial position of the IMU, \mathbf{g}^i is the inertial gravitational acceleration experienced by the IMU, and accelerometer misalignment and nonorthogonality errors have been absorbed into the measured specific force vector \mathbf{f}_m^i . It should be noted that any displacement effects caused by the gravitational acceleration acting at the CM as opposed to the IMU are neglected. Noting that the specific force is actually measured in the IMU platform frame, the translational equations

governing the dynamical evolution of the position of the IMU can be expressed as a set of first-order differential equations given by

$$\dot{\mathbf{r}}^i(t) = \mathbf{v}^i(t) \quad (17a)$$

$$\dot{\mathbf{v}}^i(t) = \mathbf{g}^i(\mathbf{r}^i(t)) + \mathbf{T}_p^i(t)\mathbf{f}_m^p(t), \quad (17b)$$

where \mathbf{v}^i is the inertial velocity of the IMU, \mathbf{f}_m^p is the sensed acceleration in the IMU platform frame, and \mathbf{T}_p^i is the attitude transformation from the platform frame to the inertial frame. Ideally, the platform frame exactly tracks the inertial frame; if this is the case, then $\mathbf{T}_p^i = \mathbf{I}_{3 \times 3}$. The continuous-time attitude representation of the platform frame evolves as [5]

$$\dot{\bar{\mathbf{q}}}_i^p(t) = \frac{1}{2}\bar{\boldsymbol{\omega}}_{p/i}^p(t) \otimes \bar{\mathbf{q}}_i^p(t), \quad (18)$$

where $\bar{\mathbf{q}}(t)$ is the pure quaternion (in right-handed, vector-first) form of the IMU frame angular velocity, defined as

$$\bar{\boldsymbol{\omega}}_{p/i}^p(t) = \begin{bmatrix} \boldsymbol{\omega}_{p/i}^p(t) \\ 0 \end{bmatrix}.$$

For notational simplicity, let $\mathbf{r}^i(t) \rightarrow \mathbf{r}(t)$, $\mathbf{v}^i(t) \rightarrow \mathbf{v}(t)$, $\boldsymbol{\omega}_{p/i}^p(t) \rightarrow \boldsymbol{\omega}_m(t)$, $\bar{\mathbf{q}}_i^p(t) \rightarrow \bar{\mathbf{q}}(t)$, $\mathbf{T}_p^i(t) = \mathbf{T}^T(t)$, and $\mathbf{f}_m^p(t) = \mathbf{f}_m(t)$, such that the equations governing the forward evolution of the position, velocity, and attitude of the space-stabilized IMU are given by collecting Eqs. (17) and (18) and making the appropriate substitutions to arrive at

$$\begin{aligned} \dot{\mathbf{r}}(t) &= \mathbf{v}(t) \\ \dot{\mathbf{v}}(t) &= \mathbf{g}(\mathbf{r}(t)) + \mathbf{T}^T(t)\mathbf{f}_m(t) \\ \dot{\bar{\mathbf{q}}}(t) &= \frac{1}{2}\bar{\boldsymbol{\omega}}_m(t) \otimes \bar{\mathbf{q}}(t). \end{aligned}$$

The continuous-time equations of motion are now discretized via analytical integration. In the following equations, it is assumed that the non-gravitational acceleration and angular velocity are constant over the time-step:

$$\mathbf{f}_{m,k} = \frac{\Delta \mathbf{v}_{m,k}}{\Delta t_k} \quad \text{and} \quad \boldsymbol{\omega}_{m,k} = \frac{\Delta \boldsymbol{\theta}_{m,k}}{\Delta t_k}.$$

The continuous time attitude evolves as

$$\dot{\bar{\mathbf{q}}}(t) = \frac{1}{2}\bar{\boldsymbol{\omega}}_m(t) \otimes \bar{\mathbf{q}}(t),$$

where $\bar{\boldsymbol{\omega}}_m(t)$ is the pure quaternion given (in right-handed, vector-first form) by

$$\bar{\boldsymbol{\omega}}_m(t) = \begin{bmatrix} \boldsymbol{\omega}_m(t) \\ 0 \end{bmatrix}.$$

Define the quaternion $\Delta \bar{\mathbf{q}}(t)$ representing the rotation from the *a priori* attitude as

$$\Delta \bar{\mathbf{q}}(t) \triangleq \bar{\mathbf{q}}(t) \otimes \bar{\mathbf{q}}_{k-1}^{-1}.$$

Solving for $\bar{\mathbf{q}}(t)$ yields

$$\bar{\mathbf{q}}(t) \triangleq \Delta \bar{\mathbf{q}}(t) \otimes \bar{\mathbf{q}}_{k-1}.$$

Assuming the time step to be small, it can be shown that the rotation vector from t_{k-1} to t_k is

$$\boldsymbol{\theta}(t_k) = \boldsymbol{\omega}_{m,k}(t_k - t_{k-1}) = \boldsymbol{\omega}_{m,k}\Delta t_k,$$

where $\Delta t_k = t_k - t_{k-1}$. Thus, the discrete quaternion propagation is given by

$$\bar{\mathbf{q}}_k = \bar{\mathbf{q}}(t_k) = \bar{\mathbf{q}}(\boldsymbol{\omega}_{m,k}\Delta t_k) \otimes \bar{\mathbf{q}}_{k-1}, \quad (19)$$

where the quaternion representation of the arbitrary rotation vector $\boldsymbol{\theta}$ is given by

$$\bar{\mathbf{q}}(\boldsymbol{\theta}) = \begin{bmatrix} \sin\left(\frac{1}{2}\theta\right) \boldsymbol{\theta}/\theta \\ \cos\left(\frac{1}{2}\theta\right) \end{bmatrix}.$$

The continuous time velocity evolves as

$$\dot{\mathbf{v}}(t) = \mathbf{g}(\mathbf{r}(t)) + \mathbf{T}^T(t)\mathbf{f}_m(t).$$

Assuming a small time step such that the non-gravitational acceleration is approximately constant, i.e. $\mathbf{a}_m(t) = \mathbf{a}_{m,k}$, such that

$$\dot{\mathbf{v}}(t) = \mathbf{g}(\mathbf{r}(t)) + \mathbf{T}^T(t)\mathbf{f}_{m,k}.$$

The first-order evolution of the rotation matrix is given by

$$\mathbf{T}(t) = \mathbf{T}(\bar{\mathbf{q}}(t)) = \mathbf{T}_{k-1} - [\boldsymbol{\theta}(t)\times]\mathbf{T}_{k-1},$$

which may be inverted to give

$$\mathbf{T}^T(t) = \mathbf{T}_{k-1}^T + \mathbf{T}_{k-1}^T[\boldsymbol{\theta}(t)\times].$$

Letting $\mathbf{r}(t) \rightarrow \mathbf{r}_{k-1}$ and defining $\mathbf{g}(\mathbf{r}(t)) = \mathbf{g}(\mathbf{r}_{k-1}) \rightarrow \mathbf{g}_{k-1}$ yields

$$\dot{\mathbf{v}}(t) = (\mathbf{T}_{k-1}^T\mathbf{f}_{m,k} + \mathbf{g}_{k-1})(t - t_{k-1}).$$

Integrating twice and once respectively, and letting $t \rightarrow t_k$ gives the following:

$$\mathbf{r}_k = \mathbf{r}_{k-1} + \mathbf{v}_{k-1}\Delta t_k + \frac{1}{2}(\mathbf{T}_{k-1}^T\mathbf{f}_{m,k} + \mathbf{g}_{k-1})\Delta t_k^2 \quad (20a)$$

$$\mathbf{v}_k = \mathbf{v}_{k-1} + (\mathbf{T}_{k-1}^T\mathbf{f}_{m,k} + \mathbf{g}_{k-1})\Delta t_k. \quad (20b)$$

In summary, the inertial position, velocity, and IMU platform attitude of a space-stabilized inertial navigation system evolve in discrete time according to Eqs. (20) and (19).

3.5 Monte Carlo Analysis

In general, any of the mechanizations of the IMU can be represented by a nonlinear dynamical system. If the relevant expressions for position (e.g. latitude, longitude and altitude or inertial Cartesian position), velocity, and attitude are collected into a state vector, \mathbf{x} , the general nonlinear dynamical system representing the forward evolution of the state for any of the IMU systems may be expressed as

$$\dot{\mathbf{x}}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{a}_k, \boldsymbol{\omega}_k, t_k), \quad (21)$$

where \mathbf{a}_k and $\boldsymbol{\omega}_k$ represent the true vehicle non-gravitational acceleration and angular velocity, respectively. For the mechanizations presented in Section 3.4, $\mathbf{f}(\cdot)$ represents one of Eqs. (13) and (7), Eqs. (15) and (16), or Eqs. (20) and (19). Functionally, the relationship between the true and measured IMU outputs is given by

$$\mathbf{a}_{m,k} = \mathbf{a}(\boldsymbol{\phi}_k, t_k) \quad \text{and} \quad \boldsymbol{\omega}_{m,k} = \boldsymbol{\omega}(\boldsymbol{\psi}_k, t_k), \quad (22)$$

where $\boldsymbol{\phi}_k$ and $\boldsymbol{\psi}_k$ represent the IMU model parameters, which also have governing evolutionary equations as

$$\dot{\boldsymbol{\phi}}_k = \mathbf{g}(\boldsymbol{\phi}_k, t_k) \quad \text{and} \quad \dot{\boldsymbol{\psi}}_k = \mathbf{h}(\boldsymbol{\psi}_k, t_k).$$

In the preceding discussions, many possibilities for the forms of \mathbf{g} , \mathbf{h} , $\boldsymbol{\phi}$, and $\boldsymbol{\psi}$ have been given. For example, for the simple additive model where the true IMU outputs are corrupted by a constant bias \mathbf{b} and a white-noise sequence \mathbf{w} , Eqs. (22) become

$$\mathbf{a}_{m,k} = \mathbf{a}_k + \mathbf{b}_a + \mathbf{w}_{a,k} \quad \text{and} \quad \boldsymbol{\omega}_{m,k} = \boldsymbol{\omega}_k + \mathbf{b}_g + \mathbf{w}_{g,k},$$

and the parameters of the IMU error model are

$$\boldsymbol{\phi}_k = \begin{bmatrix} \mathbf{b}_a \\ \mathbf{w}_{a,k} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\psi}_k = \begin{bmatrix} \mathbf{b}_g \\ \mathbf{w}_{g,k} \end{bmatrix},$$

with the governing equations $\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1}$ and $\boldsymbol{\psi}_k = \boldsymbol{\psi}_{k-1}$ (note that the white noise sequence does not require a formal time-wise evolution owing to the properties of white noise).

In order to predict the position, velocity, and attitude of the vehicle, the standard approach is to implement an approximate filter such as the extended Kalman filter (EKF). The EKF is a linear minimum mean square error filter which does not require an assumption of Gaussianity except to make statistical conclusions. In the EKF, the state estimate is represented by $\hat{\mathbf{x}}$, and the IMU outputs are used to formulate a state estimate evolution of the form

$$\hat{\mathbf{x}}_k = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{a}}_k, \hat{\boldsymbol{\omega}}_k, t_k),$$

which is accompanied by a linearized propagation of the uncertainty distribution of the states through the covariance propagation equation

$$\mathbf{P}_k = \mathbf{F}_{k-1} \mathbf{P}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_k. \quad (23)$$

In Eq. (23), \mathbf{F}_{k-1} is the tangent linear dynamics evaluated at the current estimated state and estimated IMU model parameters, and \mathbf{Q}_k is the process noise covariance matrix. It should be noted that the linearization in the EKF is performed about the current best state estimate. The two preceding equations then represent the best estimate of the vehicle state along with the uncertainty in that estimate.

Typical analysis of IMU-driven systems employs linearization about a reference truth trajectory in order to decouple the covariance propagation from the state estimate propagation; this is what is called linear covariance analysis. In doing so, the true IMU acceleration and angular velocity that are used in the forward solution to Eq. (21) are also used in the covariance propagation equation of Eq. (23), obfuscating the appearance of any influence from the IMU errors on the state estimate and covariance. Furthermore, the covariance obtained from the linear covariance analysis is interpreted as being representative of a Gaussian distribution in order to determine performance measures; however, if the distribution is not Gaussian, the calculation of performance measures can lead

to erroneous conclusions when using the assumption of Gaussianity. Additionally, the mean and covariance information from a Gaussian distribution gives no realistic representation of the closest approach of an estimate to the truth or the furthest departure of an estimate from the truth; that is, no best case or worst case scenario information can be obtained. In order to obviate erroneous statistical conclusions based on a Gaussian assumption resulting from using the covariance matrix obtained from a linear covariance analysis, Monte Carlo techniques that statistically sample the IMU error parameterization, simulate sensed IMU outputs, emulate the navigation filter in order to obtain estimates of the vehicle's position, velocity, and attitude, and then collect statistics on certain user-defined are needed.

To illustrate the Monte Carlo simulation process, consider the two dimensional transformation from polar to Cartesian coordinates, i.e.

$$\begin{bmatrix} x \\ y \end{bmatrix} = r \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}.$$

Given a distribution on $\mathbf{w} = [r \ \theta]$, it is desired to determine distribution characteristics on $\mathbf{z} = [x \ y]$. First, assume that r and θ are independent and Gaussian with mean and covariance

$$\mathbf{m}_w = \begin{bmatrix} 1 \text{ [m]} \\ 60 \text{ [deg]} \end{bmatrix} \quad \text{and} \quad \mathbf{P}_w = \begin{bmatrix} 0.02 \text{ [m]}^2 & 0 \\ 0 & 30 \text{ [deg]}^2 \end{bmatrix}.$$

Therefore, the polar to Cartesian conversion may be written as

$$\mathbf{z} = \mathbf{f}(\mathbf{w}),$$

where \mathbf{w} is the input and \mathbf{z} is the associated output. The Monte Carlo simulation process is started by drawing a number of samples from the input vector, where the sampling is performed with respect to the probability density function which governs the characteristics of the input. Once the samples are drawn, the nonlinear mapping function, \mathbf{f} , is applied to each input sample to generate a set of output samples.

For the coordinate conversion problem, the Monte Carlo simulation is initialized by drawing 1×10^5 realizations of the random vector \mathbf{w} from the distribution

$$p(\mathbf{w}; \mathbf{m}_w, \mathbf{P}_w) = |2\pi\mathbf{P}_w|^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{w} - \mathbf{m}_w)^T \mathbf{P}_w^{-1} (\mathbf{w} - \mathbf{m}_w) \right\}.$$

The set of samples drawn is shown in Fig. (12(a)). The nonlinear transformation is applied to each sample in the input space and the resulting set of Cartesian coordinates is shown in Fig. (12(b)). It is clear that the output space is not capable of being described by a Gaussian distribution; however, the Monte Carlo simulation technique can still be used to compute an accurate approximation of the mean and covariance via

$$\mathbf{m}_z = \frac{1}{N} \sum_{i=1}^N \mathbf{z}_i \quad \text{and} \quad \mathbf{P}_z = \frac{1}{N} \sum_{i=1}^N (\mathbf{z}_i - \mathbf{m}_z)(\mathbf{z}_i - \mathbf{m}_z)^T, \quad (24)$$

where

$$\mathbf{z}_i = \mathbf{f}(\mathbf{w}_i)$$

and \mathbf{w}_i is the i^{th} Monte Carlo sample. The contours representing the Gaussian distribution computed from the preceding mean and covariance equations are shown on top of the set of output-space

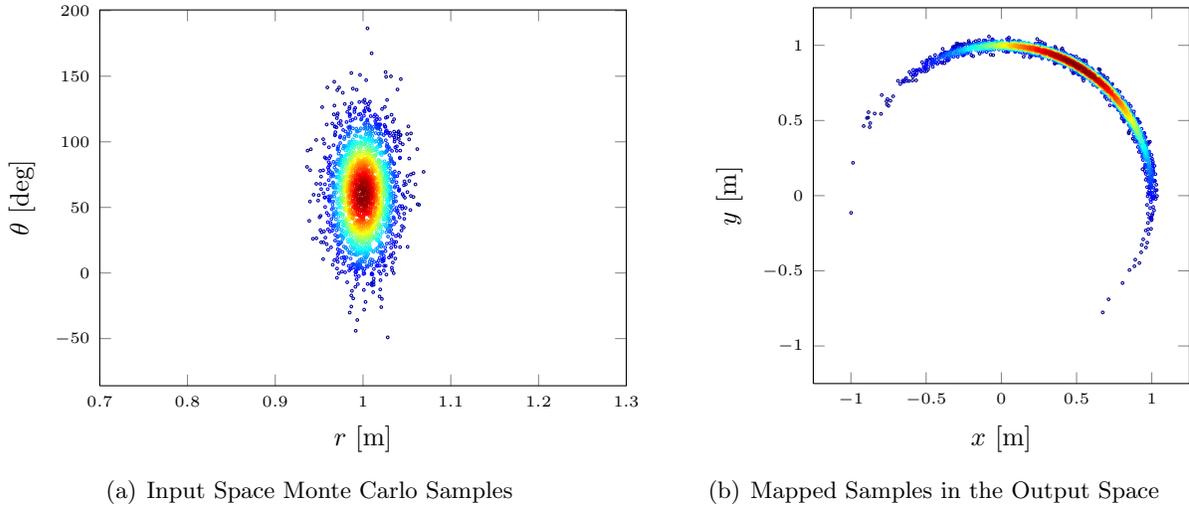


Figure 12. Input and Output Samples from the Monte Carlo Simulation

Monte Carlo samples in Fig. (13(a)). Moreover, a linearization approach can be used to map the input mean and covariance into an approximate output mean and covariance. The comparison between the linear mean and covariance and the Monte Carlo mean and covariance is shown in Fig. (13(b)), wherein it is seen that the two differ a great deal. The only characteristic that is mostly captured by the linearization process is the orientation of the covariance matrix. The centering (i.e. the mean of the distribution) is not captured accurately, and the size of the mapped covariance matrix is not captured accurately.

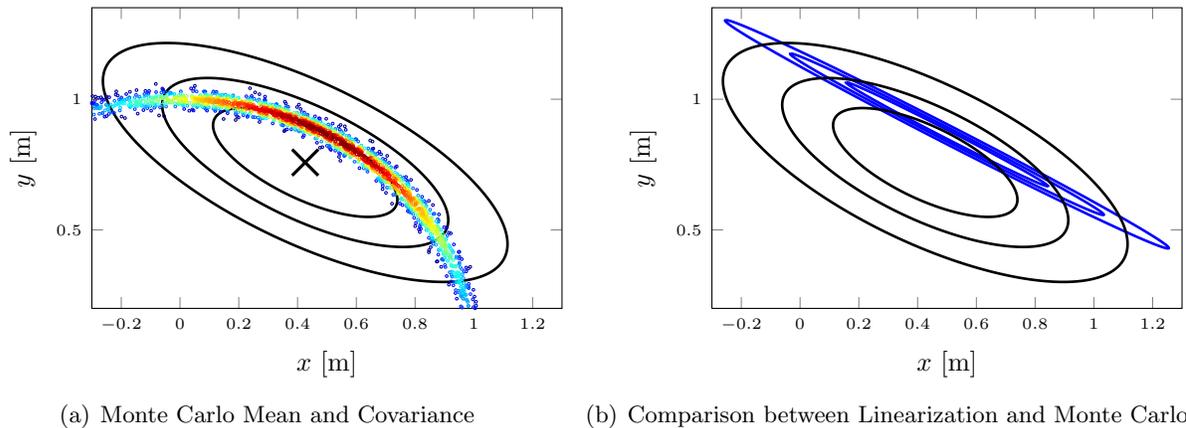


Figure 13. Distribution Contours Obtained by the Mean and Covariance from the Monte Carlo Simulation and Linearization

To further demonstrate the power of Monte Carlo simulation, consider a modification of the preceding problem. In this case, it is still assumed that the radial and angular input coordinates are independent, but now the radial direction is taken to be Gaussian distributed and the angular direction is taken to be uniformly distributed. The mean and covariance are taken to be the same as before so that the input space has the same first and second central moments as in the preceding

example. The probability density function for the input is now given by

$$p(\mathbf{w}; m_r, P_r, m_\theta, P_\theta) = |2\pi P_r|^{-1/2} \exp \left\{ -\frac{(w_r - m_r)^2}{2P_r} \right\} \cdot \frac{1}{2\sqrt{3P_\theta}}$$

$$= \frac{1}{2\sqrt{6\pi P_r P_\theta}} \exp \left\{ -\frac{(w_r - m_r)^2}{2P_r} \right\},$$

where $m_\theta - \sqrt{3P_\theta} \leq \theta \leq m_\theta + \sqrt{3P_\theta}$, such that $[m_\theta - \sqrt{3P_\theta}, m_\theta + \sqrt{3P_\theta}]$ is the support set of the uniform angular distribution. From the original specifications for the transformation problem, the means and covariances required to evaluate the input-space distribution are

$$m_r = 1 \text{ [m]}, \quad m_\theta = 60 \text{ [deg]}, \quad P_r = 0.02 \text{ [m]}^2, \quad \text{and} \quad P_\theta = 30 \text{ [deg]}^2.$$

As before, the Monte Carlo simulation is initialized by drawing 1×10^5 realizations of the random vector \mathbf{w} from the distribution $p(\mathbf{w}; m_r, P_r, m_\theta, P_\theta)$, and each sample is mapped into the output space via the nonlinear transformation. These samples are shown in Figs. (14(a)) and (14(b)), respectively. The mean and covariance is again computed from the Monte Carlo samples and

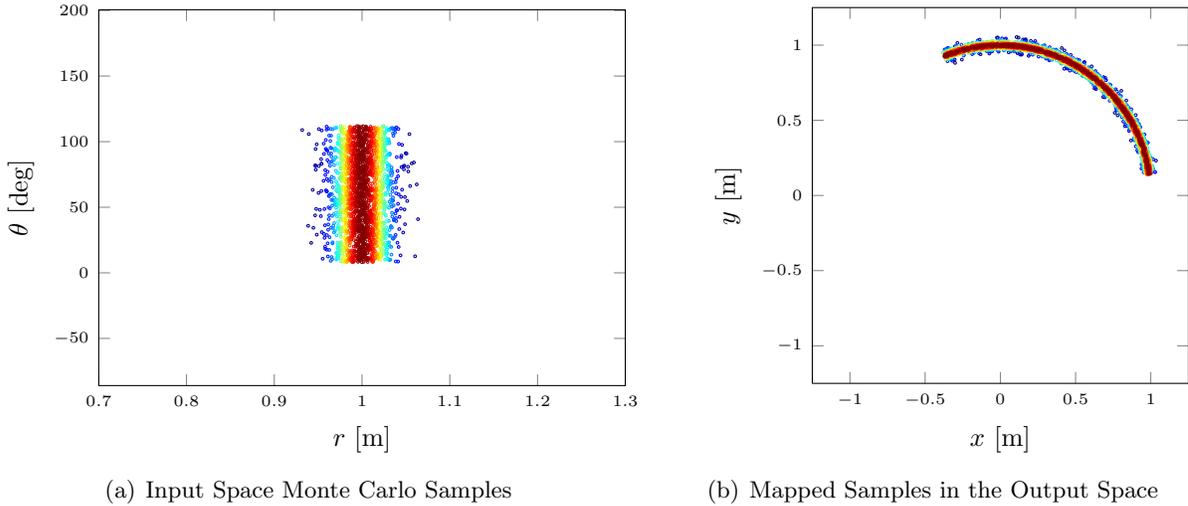


Figure 14. Input and Output Samples from the Monte Carlo Simulation (modified problem)

shown along with the output-space samples in Fig. (15(a)). Finally, a linearization approach is used to map the input-space mean and covariance into an output-space mean and covariance. The comparison between the Monte Carlo mean and covariance and the linearization mean and covariance is shown in Fig. (15(b)).

While it is clear that the linearization process once again fails to capture the true mean and covariance, it is noted that the linearized mean and covariance depicted in Fig. (15(b)) are identical to the ones depicted in Fig. (13(b)). At the same time, however, the Monte Carlo means and covariances differ between these two simulations. This occurs because the Monte Carlo simulation has and uses knowledge of the entire distribution on both the input and output spaces whereas a linearization procedure works with only the first and second central moments of the distribution. Therefore if two input distributions have the same mean and covariance, and are subjected to the same nonlinear transformation, the linearized solutions will always have the same output mean and covariance.

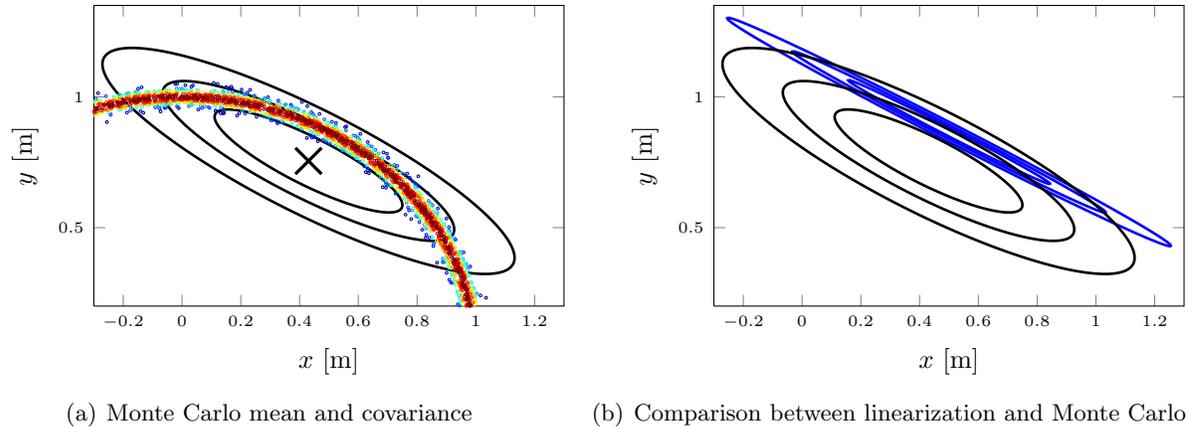


Figure 15. Distribution Contours Obtained by the Mean and Covariance from the Monte Carlo Simulation and Linearization (modified problem)

It is also worth noting that the Monte Carlo simulation enables analysis extending beyond the computation of the mean and covariance. For instance, higher-order statistical moments may be computed in a similar fashion to the mean and covariance, best case and worst case performance can be analyzed through analysis of the dispersions away from a predefined truth, or the radial distance encapsulating a given percentage of the sample points can be computed.

The problem of analyzing the performance of IMU-based navigation systems is characterized by complex, nonlinear functions where the input space is described by the IMU error sources, which may have fairly arbitrary statistical descriptions. Monte Carlo analysis naturally handles the characteristics of this problem and enables a more rich data analysis suite than is found in typical covariance-based analysis.

4.0 RESULTS AND DISCUSSION

4.1 Trajectory Simulation

In order to provide the trajectory input to SAIMUN that is outlined in Section 3.1, a ground vehicle was used to log acceleration and angular velocity data using a VectorNav VN-100² IMU. The VN-100 logged data at 40 Hz during a trajectory that took approximately 41 minutes to complete. A smoother was implemented to decrease the effect of noise on the recorded acceleration and angular velocity data. The smoothed acceleration and angular velocity data were dead-reckoned from a known starting condition to provide the true trajectory for SAIMUN to use. The dead-reckoned trajectory is not perfectly representative of the trajectory followed by the vehicle; however, for the purposes of applying SAIMUN and analyzing IMU-based navigation performance, the dead-reckoned trajectory is taken to be the true trajectory.

4.2 IMU Selection

To demonstrate the utility of SAIMUN, two inertial measurement units are compared: the Lord Microstrain 3DM-GX3-15³ and the Analog Devices ADIS16488⁴. Both IMUs utilize MEMS sensors

²<http://www.vectornav.com/Downloads/Support/PB-12-0002.pdf>, accessed 04/21/2014

³<http://files.microstrain.com/3DM-GX3-15-Inertial-Measurement-Unit.pdf>, accessed 04/21/2014

⁴http://www.analog.com/static/imported-files/data_sheets/ADIS16488.pdf, accessed 04/21/2014

to provide triaxial accelerometers, gyroscopes, magnetometers, and a pressure sensor. The IMUs are implemented to provide dead-reckoning navigation of a ground vehicle trajectory.

For this analysis, only the accelerometer and gyroscopic information is of interest. Table 3 shows the pertinent specifications for both of the chosen IMUs. These IMUs are interesting to compare due to the differences between the random walk and bias instability specifications. Table 3 shows that the ADIS16488 has a higher random walk specification and thus more white noise present in the signal. The 3DM-GX3-15, on the other hand, has a higher bias instability specification. The higher bias instability and lower random walk specifications of the 3DM-GX3-15 compared to the ADIS16488 imply that the position and attitude uncertainty associated with the 3DM-GX3-15 will grow more slowly than the ADIS16488 initially. Eventually, the position and attitude uncertainty associated with the 3DM-GX3-15 will grow larger and surpass the position and attitude uncertainty associated with the ADIS16488 as time increases.

Table 3. IMU Specifications for the ADIS16488 and 3DM-GX3-15

	Specification	3DM-GX3-15	ADIS16488
Accelerometers	Analog-to-Digital Converter Bitrate	16	32
	Range	$\pm 156.96 \frac{\text{m}}{\text{s}}$	$\pm 176.58 \frac{\text{m}}{\text{s}}$
	Scale Factor Error	$\mathcal{U}(-500, 500)$ ppm	$\mathcal{U}(-5000, 5000)$ ppm
	Axes Nonorthogonality Error	$\mathcal{U}(-103, 103)$ arcsec	$\mathcal{U}(-126, 126)$ arcsec
	Axes Misalignment Error	$\mathcal{U}(-103, 103)$ arcsec	$\mathcal{U}(-3600, 3600)$ arcsec
	Velocity Random Walk	$0.0007848 \frac{\text{m/s}}{\sqrt{\text{hr}}}$	$0.029 \frac{\text{m/s}}{\sqrt{\text{hr}}}$
	Bias Instability	$0.0003924 \frac{\text{m}}{\text{s}}$	$0.000981 \frac{\text{m}}{\text{s}}$
	Bias Instability Time	100 s	100 s
	Startup Bias	$\mathcal{U}(-0.0196, 0.0196) \frac{\text{m}}{\text{s}}$	$\mathcal{U}(-0.016, 0.016) \frac{\text{m}}{\text{s}}$
Gyroscopes	Analog-to-Digital Converter Bitrate	16	32
	Range	$\pm 600 \frac{\circ}{\text{s}}$	$\pm 450 \frac{\circ}{\text{s}}$
	Scale Factor Error	$\mathcal{U}(-500, 500)$ ppm	$\mathcal{U}(-100, 100)$ ppm
	Axes Nonorthogonality Error	$\mathcal{U}(-103, 103)$ arcsec	$\mathcal{U}(-180, 180)$ arcsec
	Axes Misalignment Error	$\mathcal{U}(-103, 103)$ arcsec	$\mathcal{U}(-3600, 3600)$ arcsec
	Angular Random Walk	$0.03 \frac{\circ}{\sqrt{\text{hr}}}$	$0.3 \frac{\circ}{\sqrt{\text{hr}}}$
	Bias Instability	$18 \frac{\circ}{\text{hr}}$	$6.25 \frac{\circ}{\text{hr}}$
	Bias Instability Time	100 s	100 s
	Startup Bias	$\mathcal{U}(-0.25, 0.25) \frac{\circ}{\text{s}}$	$\mathcal{U}(-0.2, 0.2) \frac{\circ}{\text{s}}$

4.3 Performance Metrics

In order to assess the performance of an IMU for a given trajectory, metrics need to be defined. These metrics include error probables, miss distance histograms, and standard deviation in the position, velocity, and attitude. Because the ADIS16488 and the 3DM-GX3-15 are both relatively low-end IMUs, two cases are considered: Case 1 analyzes only the first 500 seconds of the entire 41 minute trajectory while Case 2 analyzes the entire 41 minute trajectory. For each performance metric considered, the analysis is carried out for each of the two cases that are considered.

4.3.1 Error Probables. The CEP represents the radius of a circle centered on the true terminal position in the East-North plane in which the probability of landing in the circle is 0.5. The CEP is typically based on a circular bivariate normal distribution, which relies on the assumption that the terminal points of all samples will follow a Gaussian distribution. SAIMUN, however, does not use the typical Gaussian assumption to calculate the CEP. Since Monte Carlo sampling techniques are implemented, the CEP is calculated by finding the radius under which 50% of the East-North plane projection of terminal points of the Monte Carlo samples fall.

The SEP represents the radius of a sphere centered on the true terminal position in which the probability of landing in the sphere is 0.5. Again, since Monte Carlo sampling is used, no Gaussian assumption is used to calculate the SEP.

The HEP represents the altitude range centered on the true terminal position in which the probability of landing within this range, whether it is above or below the ground, is 0.5. Again, no Gaussian assumption is used to calculate the HEP due to the advantage of Monte Carlo sampling.

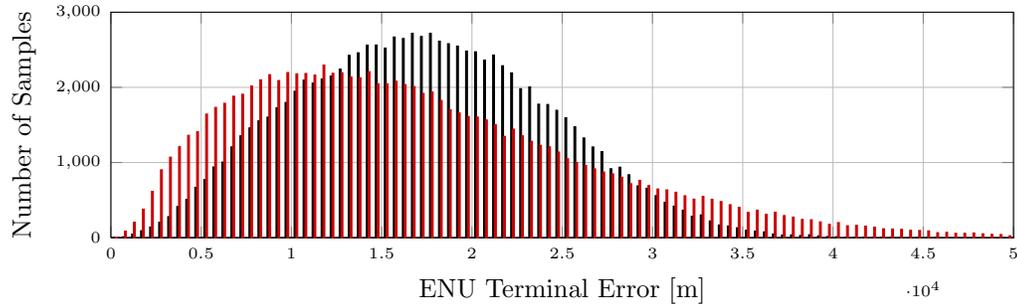
All error probables use the true trajectory at the final time to calculate the terminal position. The error probables for the ADIS16488 and the 3DM-GX3-15 are shown in Table 4 for both Case 1 and Case 2. The values of the error probable are computed by analyzing the terminal Monte Carlo samples. For example, when considering the SEP, the three-dimensional miss distance is computed for each sample. These values are then sorted, and the radius which encompasses 50% of the miss distances, which is the SEP, is determined.

Table 4. Error Probable Values for the ADIS16488 and 3DM-GX3-15

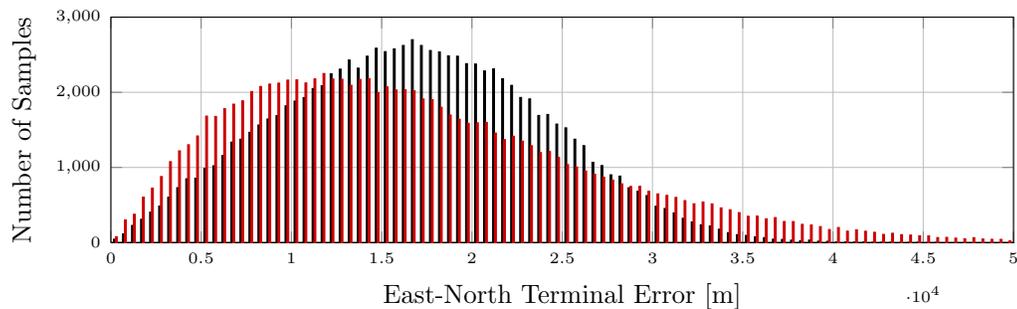
	Error Probable	MicroStrain 3DM-GX3-15	Analog Devices ADIS16488
Case 1	CEP	1.527 km	1.674 km
	SEP	1.536 km	1.719 km
	HEP	0.132 km	0.319 km
Case 2	CEP	2757 km	1000 km
	SEP	2862 km	1114 km
	HEP	665 km	364 km

4.3.2 Miss Distance and Terminal Error. The miss distance represents the magnitude of the terminal position error with respect to the propagated true trajectory at the final time. Two metrics are considered: a two-dimensional miss distance and a three-dimensional miss distance. The two-dimensional metric uses the magnitude of the East and North terminal position error components, and the three-dimensional miss distance is calculated using the magnitude of the full ENU frame error vector. By calculating the miss distance of each sample trajectory that is generated by the Monte Carlo simulation and using a histogram plot, the sample error distribution is determined. The two-dimensional and three-dimensional miss distance histograms are given in Fig. (17(a)) for the Case 1 analysis and in Fig. (17(b)) for the Case 2 analysis.

ENU projections of the terminal position error for every Monte Carlo sample are shown in Figs. (18)–(21). Figure (18) illustrates the projections for the ADIS16488 IMU as applied to the Case 1 trajectory, Fig. (19) illustrates the projections for the 3DM-GX3-15 IMU as applied to the Case 1 trajectory, Fig. (20) illustrates the projections for the ADIS16488 IMU as applied to the Case 2 trajectory, and Fig. (21) illustrates the projections for the 3DM-GX3-15 IMU as applied to the Case 2 trajectory. It is worth noting that the axis limits for the same projections



(a) East-North-Up Miss Distance Histogram

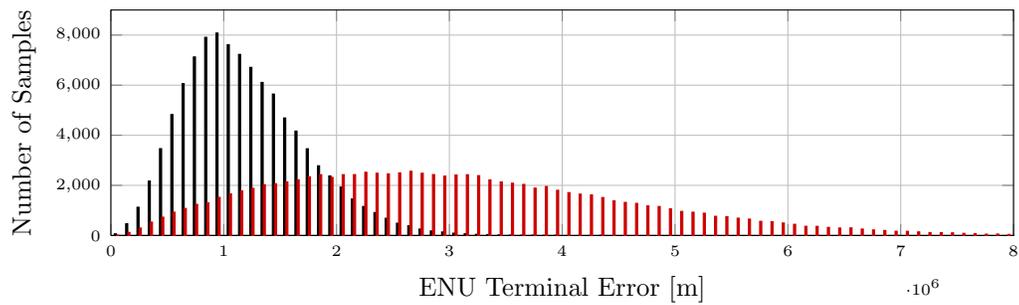


(b) East-North Miss Distance Histogram

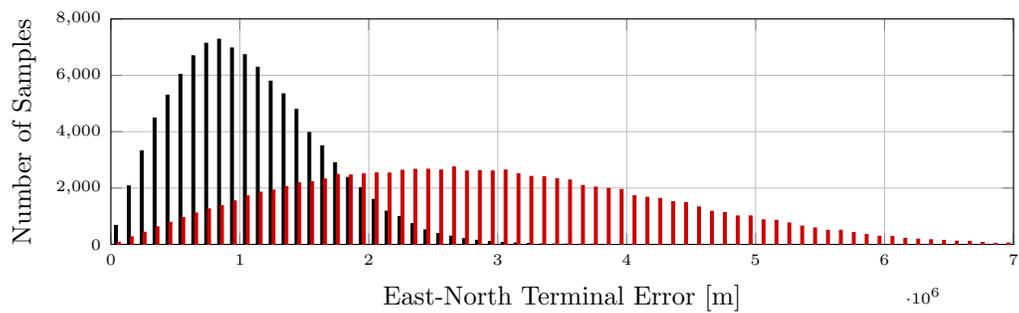
Figure 16. Miss Distance Histograms for Two- and Three-Dimensional Positions: Case 1

in Figs. (18) and (19) are identical, such that a direct comparison on the size of the uncertainty can be made. Similarly, this is true for Figs. (20) and (21). Figures (18)–(21) illustrate the non-Gaussian nature of the underlying pdf. Conventional uncertainty analysis techniques, such as linear covariance analysis, which only capture the first and second statistical moments of the underlying pdf, rely on an assumption to be made about the underlying pdf in order to calculate performance metrics such as CEP, SEP, and HEP. This assumption of the underlying pdf, usually that it is Gaussian, can lead to unrepresentative performance metrics to be calculated. Monte Carlo analysis allows the assumption of an underlying pdf to be relaxed because the mean, covariance, CEP, SEP, and HEP can be calculated from the discrete Monte Carlo samples.

4.3.3 Standard Deviation in Position and Attitude. In order to quantify uncertainty in the position and attitude states, the standard deviation, or square root of the second-central moment, is calculated and plotted as a function time based on the Monte Carlo samples (see Eqs. (24)). While beyond the scope of the current analysis, the standard deviation as computed from the Monte Carlo samples can be compared to other methods of uncertainty propagation, such as linear covariance analysis. The position and attitude standard deviations computed for the ADIS16488 and 3DM-GX3-15 IMUs can be found in Fig. (24) for the Case 1 trajectory and in Fig. (25) for the Case 2 trajectory.

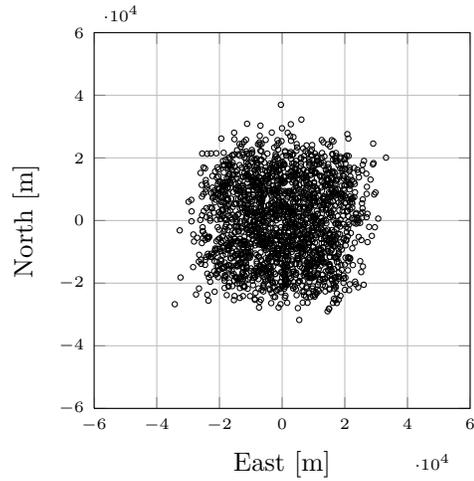


(a) East-North-Up Miss Distance Histogram

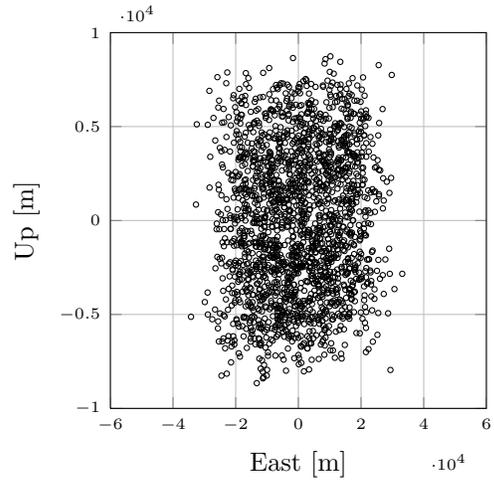


(b) East-North Miss Distance Histogram

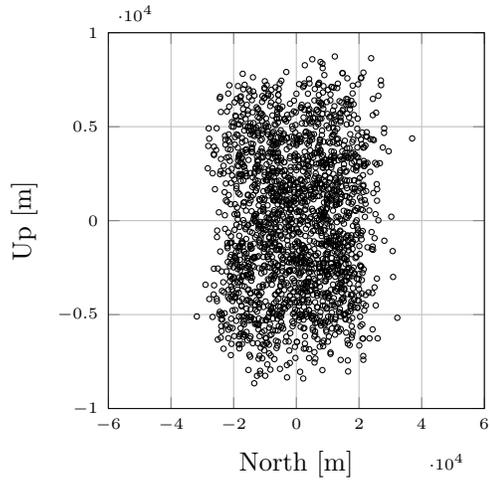
Figure 17. Miss Distance Histograms for Two- and Three-Dimensional Positions: Case 2



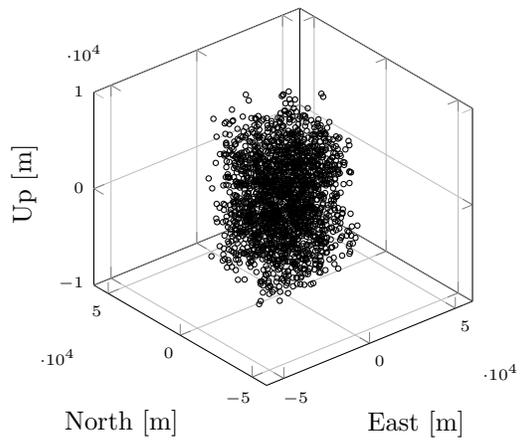
(a) East-North Projection



(b) East-Up Projection

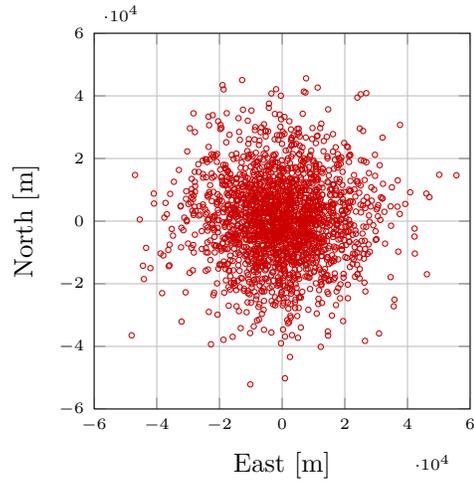


(c) North-Up Projection

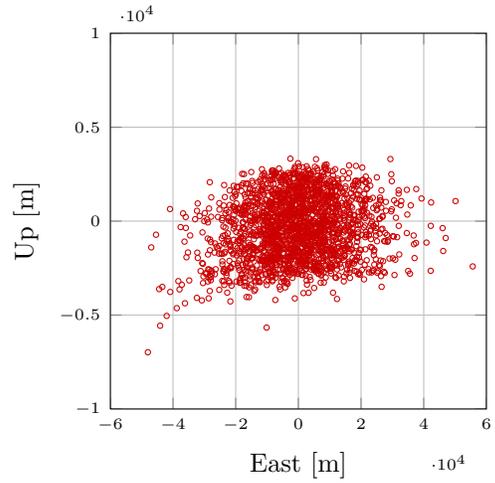


(d) Three-Dimensional View

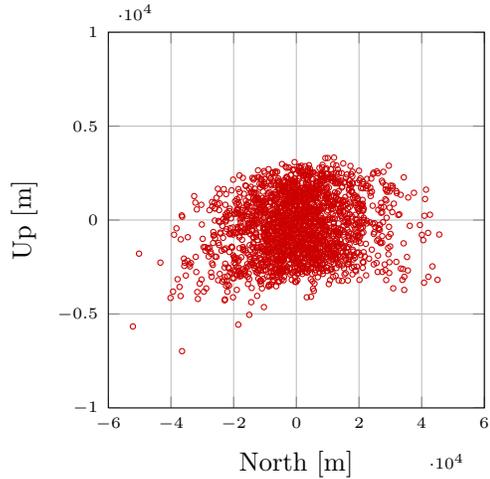
Figure 18. Terminal Position Error of the Monte Carlo Samples for the ADIS16488 IMU: Case 1



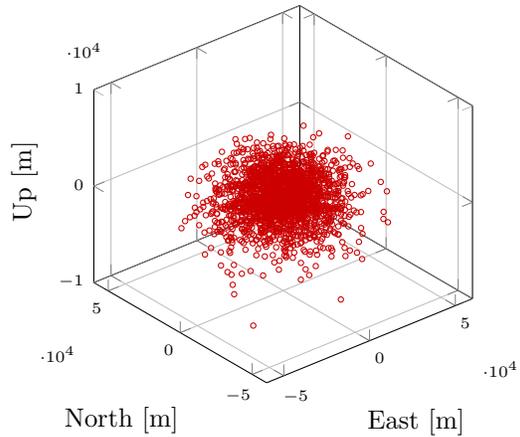
(a) East-North Projection



(b) East-Up Projection



(c) North-Up Projection



(d) Three-Dimensional View

Figure 19. Terminal Position Error of the Monte Carlo Samples for the 3DM-GX3-15 IMU: Case 12

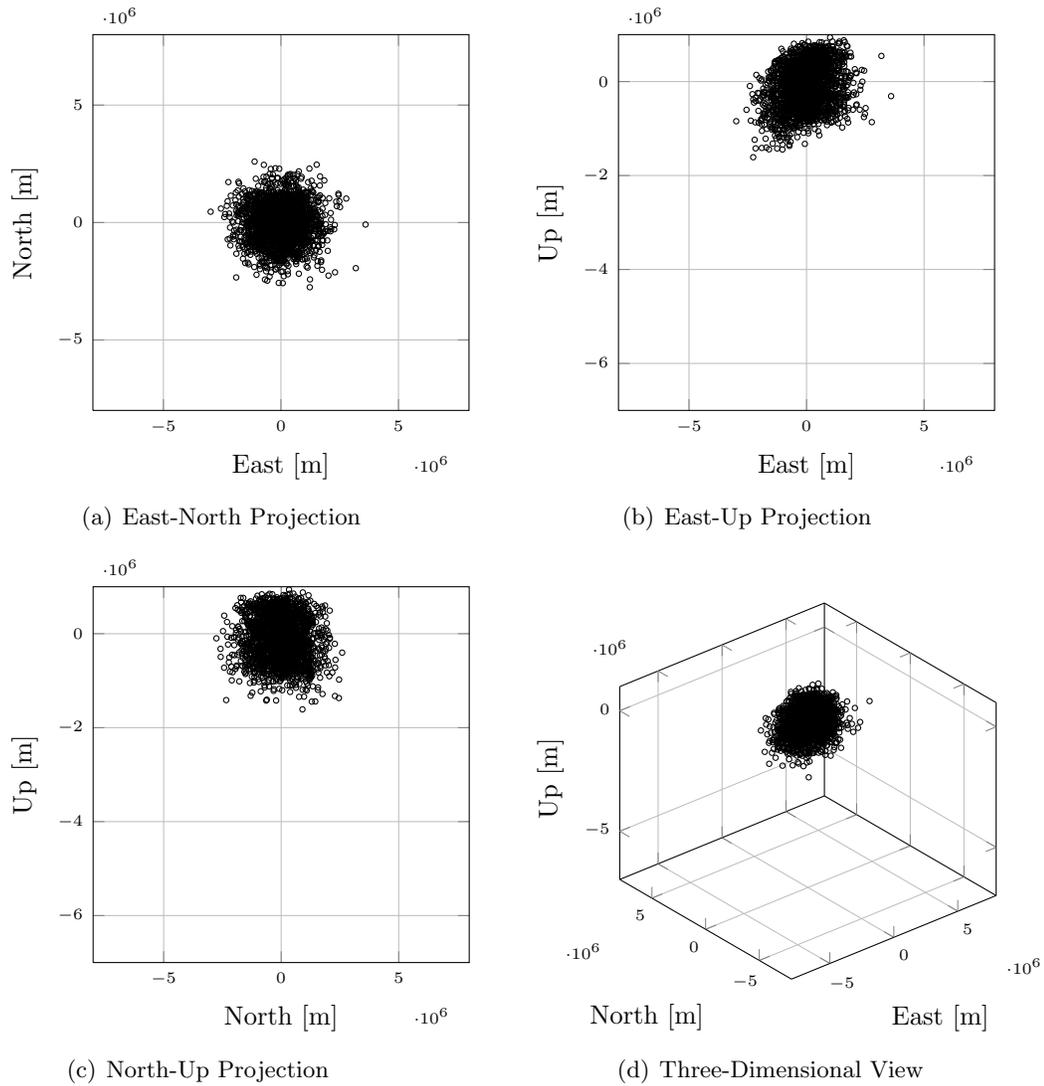


Figure 20. Terminal Position Error of the Monte Carlo Samples for the ADIS16488 IMU: Case 2

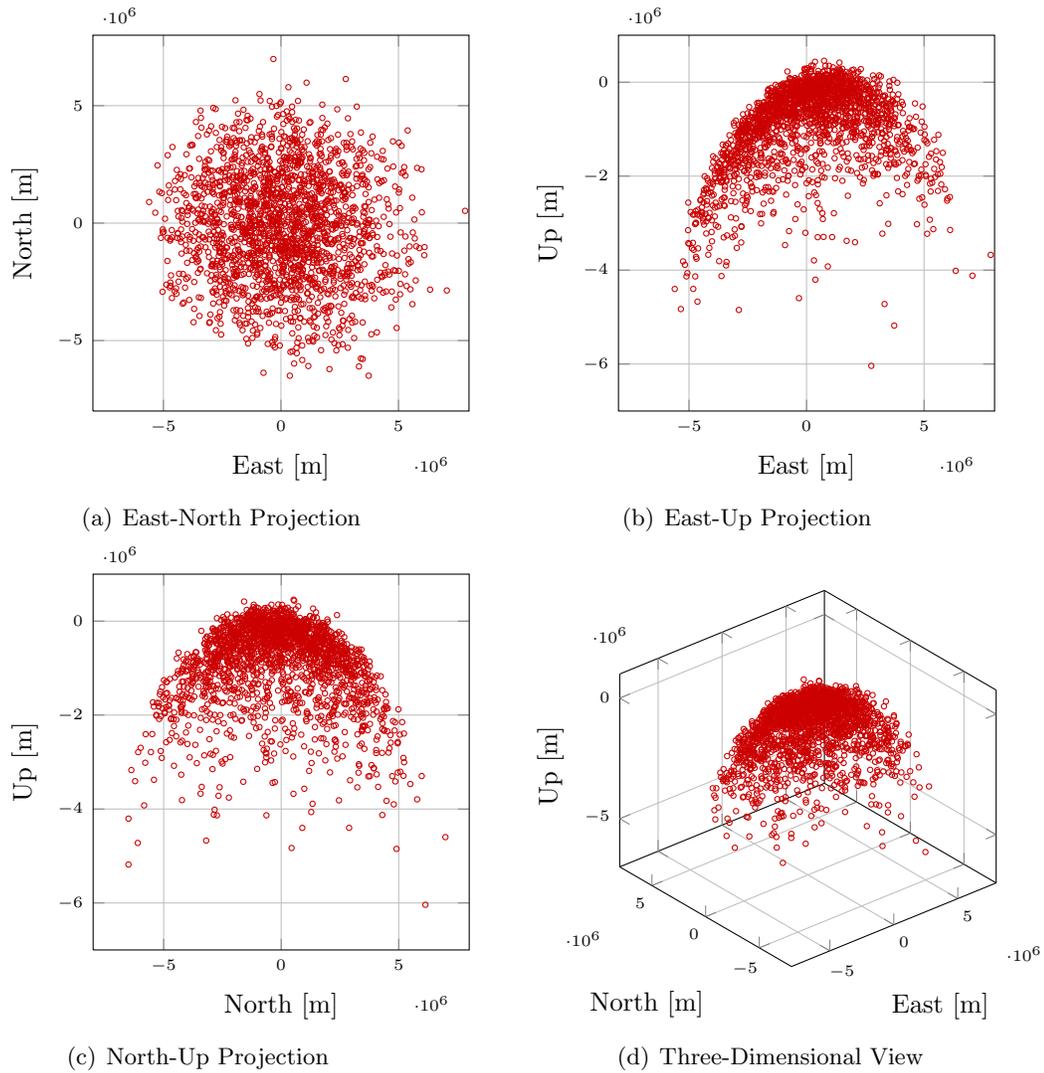


Figure 21. Terminal Position Error of the Monte Carlo Samples for the 3DM-GX3-15 IMU: Case 2

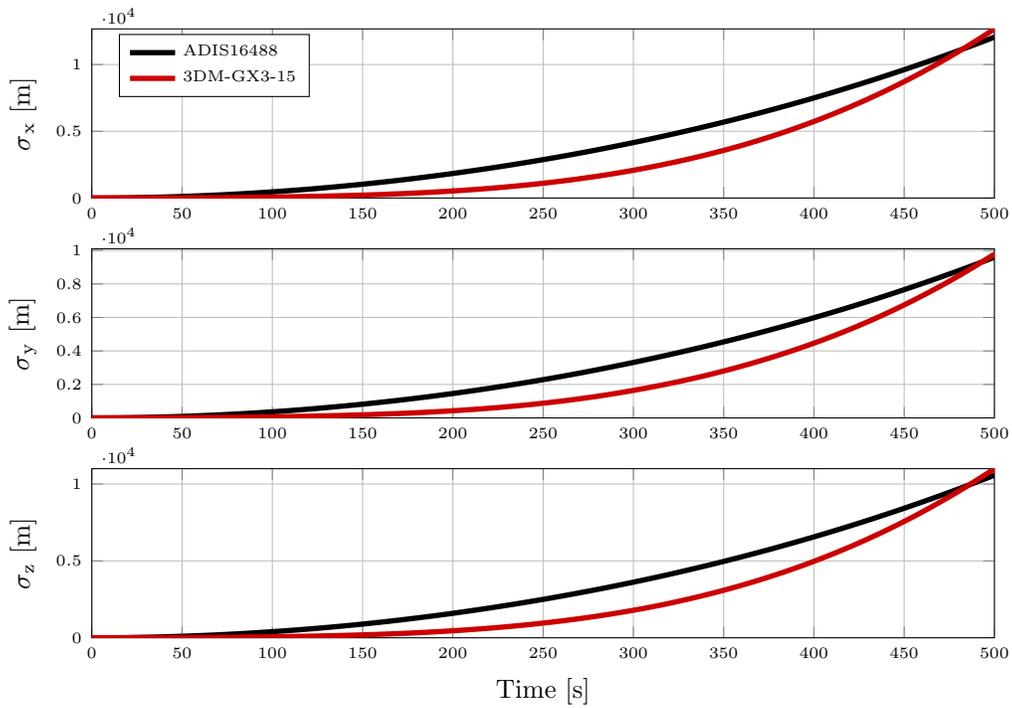


Figure 22. Position Standard Deviations as a Function of Time for the ADIS16488 and the 3DM-GX3-15 IMUs: Case 1

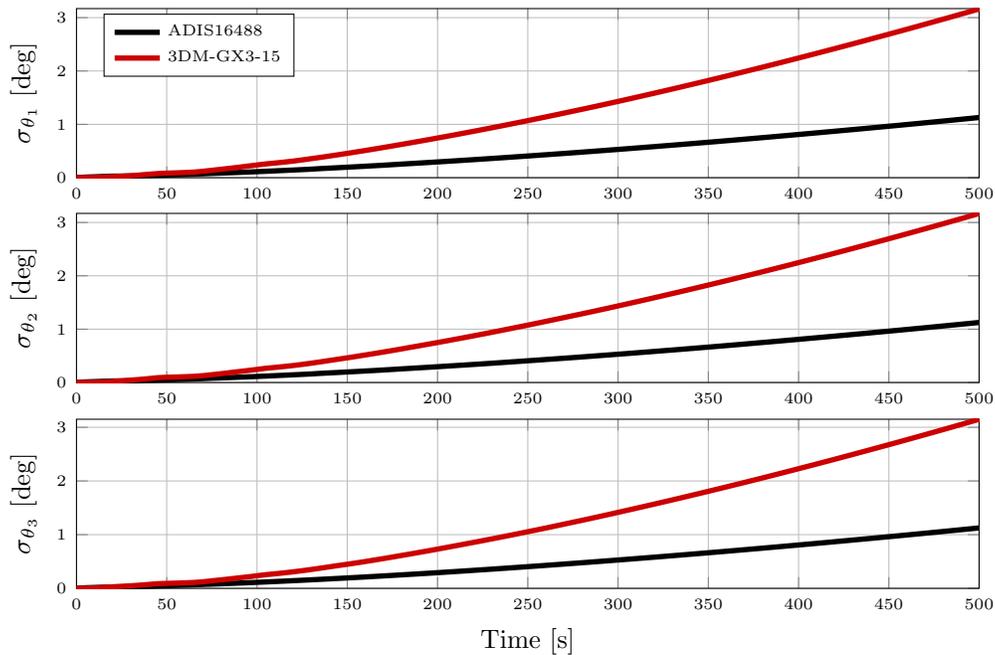


Figure 23. Attitude Standard Deviations as a Function of Time for the ADIS16488 and the 3DM-GX3-15 IMUs: Case 1

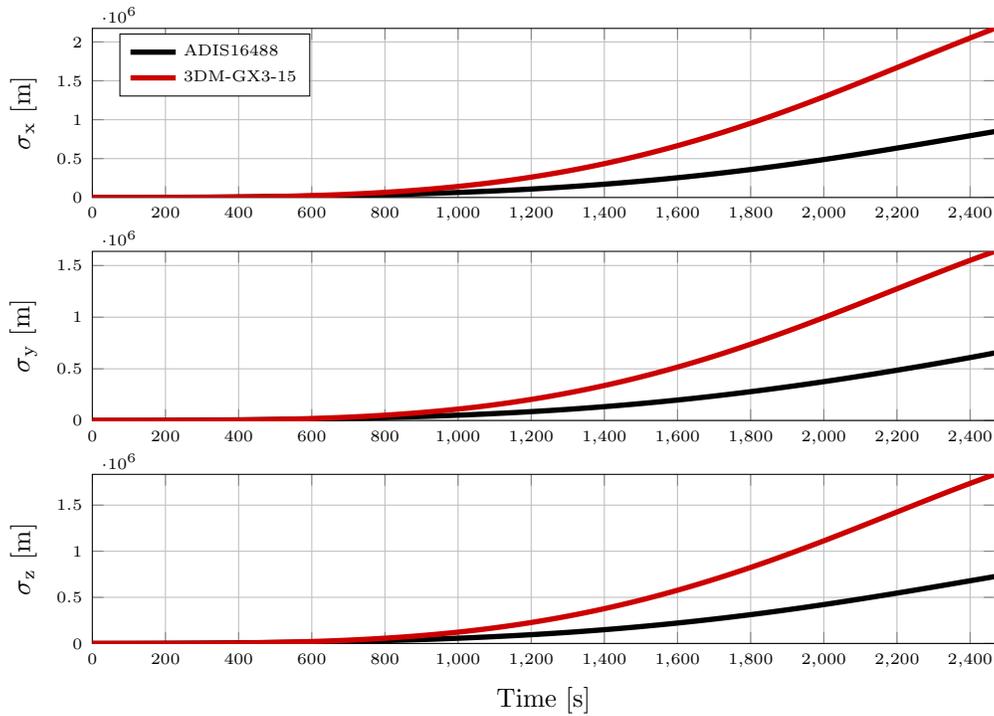


Figure 24. Position Standard Deviations as a Function of Time for the ADIS16488 and the 3DM-GX3-15 IMUs: Case 2

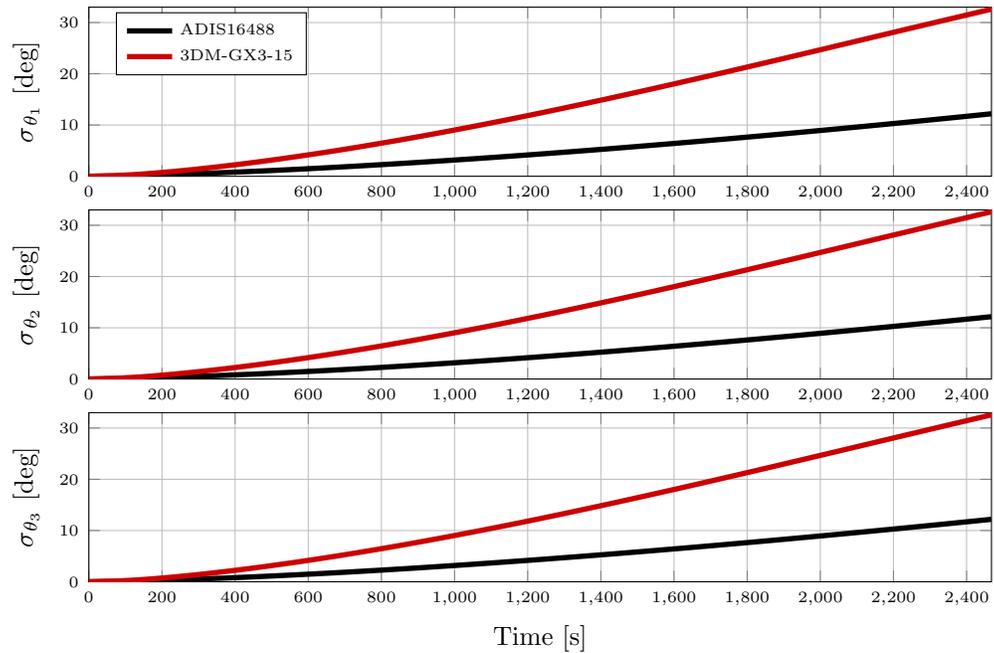


Figure 25. Attitude Standard Deviations as a Function of Time for the ADIS16488 and the 3DM-GX3-15 IMUs: Case 2

5.0 CONCLUSIONS

Before choosing an IMU to be implemented in a navigation system, SAIMUN can be used to analyze the performance of the IMU along a given trajectory or trajectories. When comparing the Lord MicroStrain 3DM-GX3-15 and the Analog Devices ADIS16488 IMUs in the navigation of an example vehicle trajectory, the lower random walk specification of the 3DM-GX3-15 causes its associated position uncertainty to grow slower during the beginning of the trajectory; however, the higher bias instability of the 3DM-GX3-15 causes its associated position uncertainty to grow faster and overtake the position uncertainty associated with the ADIS16488. For these IMUs, 480-500 seconds is observed to be the approximate window in which the performance of the IMUs is similar, as quantified by the position variance. Before this window, the 3DM-GX3-15 provides a more accurate navigation solution. After this window, the ADIS16488 provides a more accurate navigation solution. SAIMUN allows this trade study to be performed before purchasing or testing any hardware and will allow for larger trade spaces to be investigated virtually, obtaining, building, or testing any physical IMUs.

SAIMUN enables both gimbal- and strapdown-based navigation systems to be analyzed virtually in the developed simulation environment. An IMU error parameterization and a trajectory are defined in the simulation environment and Monte Carlo sample trajectories are dead-reckoned according to the appropriate dynamics for the user-defined mechanization type. SAIMUN negates the need for any assumption to be made about the underlying distribution of the navigation solution to calculate performance metrics, such as is necessary in the more traditional linear covariance analysis. The Monte Carlo analysis implemented in SAIMUN can more accurately quantify desired performance metrics, mean, and covariance, without requiring any underlying assumptions on the distribution of the state uncertainties to be made.

Currently, SAIMUN is capable of rendering navigation solutions for gimbal- and strapdown-based navigation systems, while considering IMU error sources due to startup bias, bias instability, thermo-mechanical zero-mean white noise, scale factor errors, axes misalignment errors, axes nonorthogonality errors, and quantization effects caused by analog-to-digital conversion. Where appropriate, SAIMUN allows for a selection from multiple probabilistic representations of the IMU error sources, such as Dirac, Gaussian, and uniform distributions. The structure of SAIMUN supports future extensions, such that additional IMU mechanizations, IMU parameter errors, and error source distributions can be incorporated. On the analysis side of SAIMUN, this is also true; that is, analyses extending beyond those presented in this report can be incorporated. The simulation and analysis environment is designed to be modular such that these modifications are straightforward to implement within SAIMUN.

REFERENCES

- [1] Woodman, Oliver J., *An Introduction to Inertial Navigation*, Tech. Rep. UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, August 2007.
- [2] Flenniken, Warren S., *Modeling Inertial Measurement Units and Analyzing the Effect of their Errors in Navigation Applications*, Master's thesis, Auburn University, Auburn, AL, December 2005.
- [3] "IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros," *IEEE Std 952-1997*, 1998, pp. i-77.
- [4] Wang, H. G. and Williams, T. C., "Strategic Inertial Navigation Systems," *IEEE Control Systems Magazine*, Vol. **28**, No. 1, February 2008, pp. 65 – 85.
- [5] DeMars, Kyle J., *Precision Navigation for Lunar Descent and Landing*, Master's thesis, The University of Texas at Austin, Austin, TX, May 2007.
- [6] Zanetti, Renato, *Advanced Navigation Algorithms for Precision Landing*, Ph.D. thesis, The University of Texas at Austin, Austin, TX, December 2007.
- [7] Britting, Kenneth R., **Inertial Navigation Systems Analysis**, Wiley Interscience, New York, 1st ed., 1971.
- [8] National Imagery and Mapping Agency, *Department of Defense World Geodetic System 1984, Its Definition and Relationships With Local Geodetic Systems*, Tech. Rep. TR8350.2, St. Louis, MO, 3rd ed., January 2000.

APPENDIX - CALCULATING ALLAN VARIANCE

The Allan deviation plot of generic process $\omega(t)$, which is sampled at a constant frequency f_s , can be produced by the following algorithm:

1. Log a number of samples, N , from the generic process $\omega(t)$

$$N = f_s T,$$

where T is duration of the data series.

2. Create a vector of monotonically increasing averaging times, defined as

$$t_{\text{av}} = [t_1 \quad t_2 \quad \dots \quad t_n],$$

where $t_1 \geq 1/f_s$ and $t_n \leq T/2$.

3. Divide $\omega(t)$ into M_i clusters for the i^{th} t_{av} as

$$M_i = \left\lfloor \frac{T}{t_{\text{av},i}} \right\rfloor,$$

where $i = 1, 2, \dots, n$ and the $\lfloor \cdot \rfloor$ function represents rounding towards negative infinity.

4. For each cluster M_i , compute the time average of $\omega(t)$ for each k^{th} portion of cluster size M_i

$$\bar{\omega}_{i,k} = \frac{1}{f_s t_{\text{av},i}} \sum_{j=f_s t_{\text{av},i}(k-1)+1}^{f_s t_{\text{av},i} k} \omega(j),$$

where $k = 1, 2, \dots, M_i$

5. The Allan variance for the i^{th} averaging time, $\sigma^2(t_{\text{av},i})$ is then calculated as

$$\sigma^2(t_{\text{av},i}) = \frac{1}{2(M_i - 1)} \sum_{k=1}^{M_i-1} (\bar{\omega}_{i,k+1} - \bar{\omega}_{i,k})^2.$$

6. The Allan deviation, $\sigma(t_{\text{av},i})$, is then plotted as a function of the averaging time, $t_{\text{av},i}$, to produce the Allan deviation plot.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

AIRS	Advanced Inertial Reference Sphere
CEP	Circular Error Probable
CM	Center of Mass
EKF	Extended Kalman Filter
ENU	East-North-Up
HEP	Height Error Probable
IMU	Inertial Measurement Unit
MEMS	MicroElectricalMechanical Systems
NED	North-East-Down
pdf	Probability Density Function
SAIMUN	Simulation and Analysis of IMU-based Navigation
SEP	Spherical Error Probable
UT	Universal Time
UT1	Universal Time corrected for polar wandering
UTC	Coordinated Universal Time
VN	VectorNav
VRW	Velocity Random Walk
WGS-84	1984 World Geodetic System

DISTRIBUTION LIST

DTIC/OCP 8725 John J. Kingman Rd, Suite 0944 Ft Belvoir, VA 22060-6218	1 cy
AFRL/RVIL Kirtland AFB, NM 87117-5776	2 cys
Official Record Copy AFRL/RVSV/Morgan Baldwin	1 cy